AD 740605

RADC-TR-72-2, Volume I
Final Technical Report
January 1972

RASTER PLOTTER SOFTWARE
Volume I Technical Report

United Aircraft Corporation

Approved for public release;
distribution unlimited.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) United Aircraft Corporation Hamilton Standard Division Windsor Locks, Connecticut 06096	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

RASTER PLOTTER SOFTWARE
Volume I - Technical Report

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report

5. AUTHOR(S) (First name, middle initial, last name)

Rodney T. Boorse
David L. Sharp
Gerald L. Sperber

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
January 1972	72	0

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
F30602-71-C-0107	HSER 5924
Job Order No. 60350000	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-72-2 Volume I (of two)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Rome Air Development Center (IRAG) Griffiss Air Force Base, New York 13440

13. ABSTRACT

A color coding scheme for hierarchically representing the cartographic symbology associated with JOG series 1501-A charts has been developed and implemented at the RADC Experimental Cartographic Facility (ECF). The scheme employs a three level hierarchy, with up to ten colors available for each level of the hierarchy. The first level of color coding is accomplished on a registered, transparent overlay to the base manuscript. The annotative procedure is straightforward and permits rapid color tagging with very low error raster.

Based on this color coding scheme, computer programs which accomplish the necessary recognition and interpretation of these color codes were developed and demonstrated. Color tagged raster data recorded by the RADC Automatic Color Scanning Device comprised the input to these programs, while the output comprised lineal data fully annotated in accordance with the prescribed code structure. In this process, separate scannings are required for the base manufacture and the coded overlay.

In addition, computer programs which symbolize pre-formatted raster data for output on the RADC Graphic Plotter were developed and demonstrated. These programs accept edited and symbolized raster line center data, which has been sorted by the assigned final printed graphic color, and generated the aperture and density level commands required to drive the Graphic Plotter.

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Automated Cartography Computer Programming Raster Data Processing Color Coding						

SAC -Griffiss AFB NY

UNCLASSIFIED
Security Classification

RASTER PLOTTER SOFTWARE
Volume I - Technical Report

Rodney T. Boorse
David L. Sharp
Gerald L. Sperber

United Aircraft Corporation

Approved for public release;
distribution unlimited.

FOREWORD

This report describes the work performed by the Hamilton Standard Division of United Aircraft Corporation, Windsor Locks, Connecticut, in accordance with the terms and requirements of contract F30602-71-C-0107, Job Order Number 60350000, for Rome Air Development Center, Griffiss Air Force Base, New York. William G. McLellan (IRAG) was the RADC Project Engineer administering the program. Secondary report number is HSER 5924.

In addition to the authors, Hamilton Standard personnel contributing to program implementation and the preparation of this report were Messrs. J.R. Byrd, R.G. Hubbard, H.R. Lee, and R.I. Skolnick.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

William G McLellan

Approved: WILLIAM G. MCLELLAN
Project Engineer

Franz H Dettmer

Approved: FRANZ H. DETTMER
Colonel, USAF
Chief, Intel & Recon Division

FOR THE COMMANDER: Fred I Diamond
FRED I DIAMOND
Acting Chief, Plans Office

ABSTRACT

A color coding scheme for hierarchially representing the cartographic symbology associated with JOG Series 1501-A charts has been developed and implemented at the RADC Experimental Cartographic Facility (ECF). This scheme employs a three level hierarchy, with up to ten (10) colors available for coding each level of the hierarchy. The first level of color coding is accomplished directly on the cartographic base manuscript in a manner resembling current compilation procedures, while the second and third levels of coding are accomplished on a registered, transparent overlay to the base manuscript. The annotative procedure is straightforward and permits rapid color tagging with very low error rates.

Based on this color coding scheme, computer programs which accomplish the necessary recognition and interpretation of these color codes were developed and demonstrated. Color-tagged raster data recorded by the RADC Automatic Color Scanner Device (ACSD) comprised the input to these programs, while the output comprised lineal data fully annotated in accordance with the prescribed code structure. In this process, separate scannings are required for the base manuscript and the coded overlay.

In addition, computer programs which symbolize pre-formatted raster data for output on the RADC Graphic Plotter were developed and demonstrated. These programs accept edited and symbolized raster line center data which has been sorted by the assigned final printed graphics color, and generate the aperture and density level commands required to drive the Graphic Plotter.

EVALUATION MEMO

This report documents one of a continuing series of efforts aimed at the development of an Advanced Cartographic System (ACS) for the Aeronautical Chart and Information Center (ACIC). The first objective of the effort was to develop a hierarchical color coding scheme for use with the Automatic Color Scanning Device (ACSD). The scheme developed involves the use of three colors and allows approximately three hundred features to be recognized. The first color is applied directly to the feature on the base manuscript. The remaining two colors are placed on an overlay which is registered to the base manuscript. These two colors intersect the feature they identify. The scheme works well, however, it requires that the feature be converted to a lineal format in a continuous manner. The current raster to lineal conversion process produces a segmented representation of the feature and thus separates the two overlay colors from all but the specific segment they are associated with. There are two possible solutions to this problem. The first is to tag all of the segments. This is a cumbersome and time consuming task. The second is to improve the linealization process to the point where it does produce a continuous record.

The second objective of the effort was to implement a computer program for applying "symbology" to edited raster data for output on the Graphic Plotter. The "symbology" in this case is represented by spot sizes equivalent to the desired line weights on the final products. The program runs on the DEC PDP-9 and assigns the proper aperture and density values to the raster data. These assignments can be made automatically to all or some of the data based on a table contained within the program. Optically, the user can override the program and assign specific printing information to any or all of the raster data depending on his particular requirements.

William G. McLellan
WILLIAM G. MCLELLAN
Project Engineer

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

1. BACKGROUND

a. The Experimental Cartographic Facility

RADC is presently engaged in a research and development program to design and develop hardware and application software programs for an Advanced Cartographic System (ACS) to be deployed at the Aeronautical Chart and Information Center (ACIC). Extensive government and private industry expertise are being expended to establish a realistic, yet flexible system design philosophy consistent with the existing and projected aeronautical chart finishing requirements of the ACIC. As part of its continuing development program, RADC has assembled at its Experimental Cartographic Facility (ECF) a wide variety of computer and cartographic-oriented hardware units. These include commercially available computers, storage devices, and peripheral input-output units, and a complement of commercial and experimental digitizing and plotting equipment.

b. The Chart Finishing Subsystem

Using these various equipments, one of the major efforts to date has been to configure a system for automating the chart finishing operation which produces the final color separation negatives for chart production. An essential function in automating this process has been to devise a means for rapidly and accurately converting hand drawn compilation manuscripts into digital form for computer processing. This processing is required to separate the data by its identifying feature color. edit that data to the quality level required by the finished manuscript, and assign the required lineweight symbolization for final plotting.

c. Raster Processing Subsystem

(1) Raster Data Input

Since compilation manuscripts are customarily drawn in colored leads, or pencils, one of the experimental devices developed to perform this conversion was the Automatic Color Scanning Device (ACSD). This is an automatically operated raster scanning equipment which senses and records both positional location and color identity of raster scanned data. In the current equipment, a maximum of ten (10) identifiable colors can be discriminated, which is adequate for the color coding and subsequent separation of the data by feature color class.

(2) Raster Data Output

Because the final output symbolization of lineal features is by lineweight as well as by color, another of the experimental devices developed was the Graphics Plotter. This is an automatically operated raster plotting equipment which generates the required linewidths directly from linecenter data by directing a laser beam onto photographic film through an aperture whose diameter corresponds to the lineweight desired. In the current equipment, a maximum of four (4) different lineweights (apertures) may be selected and plotted in this manner during a single pass.

2. PROGRAM OBJECTIVES

The objectives of this program were to expand the capabilities of the current Raster Processing Subsystem (RPS) by:

- Developing and implementing a color coding scheme for the raster data recorded by the ACSD. The purpose of this scheme would be to extend the present feature class identification capability to the level necessary to hierarchially identify all classes, subclasses, and types of features as found on JOG 1501-A series charts.

- Developing and implementing software for symbolizing raster data for output on the Graphic Plotter. The purpose of these programs would be to convert existing line center formatted raster data to the specific lineweight and density command codes required to operate that plotting device.

SECTION II

COLOR CODING STUDY AND INVESTIGATION

1. FEATURE IDENTIFICATION REQUIREMENT

The Automatic Color Scanning Device (ACSD) currently in operation at the RADC ECF constitutes a very rapid and accurate means for converting hand drawn manuscripts to digital form, but with its constraint of ten (10) identifiable colors it can accomplish automatic feature identification only to the class level. The feature identification requirement established by the Contract Statement of Work (SOW) was that "A way of color coding the feature data must be developed which allows recognition and identification of the individual feature types". This meant that the basic machine identification capability had to be expanded by between one and two orders of magnitude.

It was further required that "To the extent possible, the color code used shall have a hierarchial structure. That is, the appearance of a particular color, e.g. blue, as the first color in a sequence should indicate that a particular class of feature, e.g. drainage, follows". Both the magnitude and hierarchy of the code structure were to be determined by analyzing the specifications for the JOG Series 1501-A charts.

Finally, when determining the actual means of implementing the resultant coding scheme, due consideration had to be given to its impact on the current base manuscript preparation process, present ACSD scanning procedures and performance, and existing post-processing software. A full definition of these interrelated problem areas is provided in Paragraph 2.

2. PROBLEM DEFINITION

a. Code Structure

The basic requirement of the color code structure was that it be capable of hierarchially identifying the individual feature types set forth in the ACIC Specification for JOG Series 1501-A Charts. An in-depth examination and categorization of the actual symbology contained within that specification was performed during the initial part of the study phase. The reason for this was to define and resolve questions relating to the code structure at the level which matters most to the eventual cartographic user. The principal areas which impact this code structure are discussed below.

(1) Feature Itemization and Classification

A tabulation of the individual feature types listed in Appendix I, Symbolization, of the JOG Series 1501-A Specification yields a total of 242 types

of features, organized into the six (6) classes as shown in Table I.

TABLE I

JOG SERIES 1501-A FEATURE CLASSIFICATION

Feature Class	Number of Feature Types
1. Drainage	50
2. Relief	57
3. Culture	62
4. Roads	29
5. Coastal Hydrography	27
6. Aeronautical	17
Total	242

In addition to class and type, a third level of classification, termed sub-class, is sometimes employed. Therefore, a three-level hierarchy constituting feature class, sub-class, and type identification was seen to fulfil the normal cartographic classification structure requirements.

While the class and type groupings are clearly defined in the specification, the division of classes, and the grouping of types, into subclasses is not specifically stated. For study and investigation purposes. however, it was necessary to consider such groupings in order to observe how the hierarchy might be affected. Tables II through V show what appeared to be reasonable subclass groupings for the largest four feature classes. As a possible additional, or alternate, level of classification, these tables also show type groupings. Within each subclass, these type groupings represent those types which are characterized by the same output symbolization. For the particular groupings shown, the subdivision of classes into subclasses ranges from seven to nine, of subclasses into type groups from one to nine, and from subclasses into individual types from one to twelve. Presuming a maximum of ten (10) ACSD recognizable colors within any single level of the code structure, it is seen that the latter grouping exceeds the code-level limit of ten, but does so only in two instances (drainage: natural areas and shoreline), so that just a minor regrouping would be needed to adapt the desired hierarchy to the available code structure.

The basic conclusion, therefore, is that a three level code, with a limit of ten (10) groupings per level, is essentially adequate for constructing a hierarchy based on the specified JOG Series 1501-A feature type classification.

JOG#	SUBCLASS-TYPE
401	10-1
402	10-2
403	10-3
404	10-4
405	10-5
406	Symbol
407	Symbol
408	Symbol
409	Symbol
410	Symbol
411	10-6
412	10-6
413	Symbol
414	5-1
415	10-8
416	5-1,2,3,4
417	4-1
418	Culture 10-[illegible]
419	Culture 10-2
420	Culture 10-3
421	4-1
422	[illegible]-1
423	4-2
424	4-3
425	8-1
425A	8-2,3,4,
426	2-1
427	8-5
428	8-6
429	8-7
430	7-6
431	3-1
432	6-2
433	6-3
434	6-4
435	9-1
436	9-2
437	9-3
438	9-4
439	9-5
440	9-6
441	6-1
442	6-1
443	9-7
444	6-5
445	6-1
446	3-2
447	Symbol
448	10-9
449	Symbol

JOG#	SUBCLASS-TYPE
501	7-1
502	7-2
503	7-3
504	Drainage 9-3
505	Drainage 9-3
506	7-4
507	7-5

CLASS - RELIEF

SUBCLASS	COLOR	TYPE GROUP 1	2	3	4	5	6	7	8	9
Land Features	10	Index Contour	Intermediate Contour	Supp'ement Contour	Index Approx. Contour	Intermediate Approx. Contour	Form Lines	Hachures	Fault Scarp	Mountain Pass
Snow-Ice Features	9	Index Contour	Intermediate Contour	Supplement Contour	Index Approx. Contour	Intermediate Approx. Contour	Form Lines	Hachures	U	
Sand and Beach Perimeter *	8	Sand Area	Dunes. Cresent	Dunes: Lateral	Dunes: Star	Beach, Sandy	Beach Gravelly	Beach, Boulder Covered		
Area Perimeters *	7	Woods, Brushwood, Jungle	Orchards, Plantations, Vineyards	Tropical Grass	Tundra	Scattered Trees	Distorted Surface Area			
Glacier, and Ice	6	Perimeter * of Glacier, Ice Shelf, Ice Pack	Glacier, Approx. Contours	Glacier Form Lines	Glacier Moraine Perimeter	Ice Cliff *				
Large Cliffs * Depressions	5	Cliff Lowest Depression Contour	Second Depression Contour	Third Depression Contour	Highest Depression Contour					
Cut and Fill, * Small Cliffs	4	Cliff, Cut, or Fill less than Contour Internal	Cut Exceeding Contour Internal	Fill Exceeding Contour Internal						
Miscellaneous	3	Strip Mines Tailing Piles Mine Dumps	Asphalt Lake							
Miscellaneous	2	Sand in Non-Tidal Waters (perimeter)								

* - Indicates Areal Data

U - Unassigned Code

TABLE II. SAMPLE CLASSIFICATION FOR RELIEF

JOG #	SUBCLASS-TYPE
30[illegible]	[illegible]-1
30[illegible]	8-5
30[illegible]	[illegible]-2
304	8-1
305	8-2
306	8-3
307	8-4
308	10-1
309	8-1
31[illegible]	10-2
3[illegible]	[illegible]-2
[illegible]	10-3
3[illegible]	[illegible]-4
[illegible]14	[illegible]-4
315	[illegible]-5
3[illegible]	4-1
3[illegible]	3-[illegible]
3[illegible]	4-[illegible]
31[illegible]	3-2
319A	10-[illegible]
320	10-6
321	[illegible]-7
[illegible]	[illegible]-1
[illegible]	[illegible]-2
324	6-3
325	6-4
326	Point Feature
327	[illegible]-8
[illegible]	5-1
[illegible]	5-2
[illegible]	5-3
331	7-1, 7-2
332	7-1, 7-2
333	7-3, 7-4
[illegible]	[illegible], 7-[illegible]
[illegible]5	9-[illegible]
33[illegible]	9-[illegible]
3[illegible]	9-[illegible]
33[illegible]	[illegible]
[illegible]	[illegible]-2
34[illegible]	[illegible]
34[illegible]	7-5, 7-6
34[illegible]	[illegible]
34[illegible]	[illegible]-4
[illegible]	[illegible]-1
[illegible]	Point Feature
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

CLASS - DRAINAGE

SUBCLASS	COLOR NO.	TYPE GROUP 1	2	3	4	5	6	*7	8	9
Natural Areas *	[illegible]	Marsh 335 336 337 338	Mangrove 339 509	Nipa 3[illegible] [illegible]05	Land subject to inundation	Wadi 346 Sebkra 347	Wet Sand 348	U	U	
Single Line Streams, Canals, etc.	1	Perennial Surveyed	Perennial, Unsurveyed	Intermittent or Dry	Unclassified	Disappearing	Navigable Canal and Stream 320	Abandoned Canal 321	Canal, Conduit, Ditch 327	U
Shoreline * Open Water Lakes, Ponds, Reservoir, Double Line Streams	8	Definite 301 304 309 334	Unsurveyed 303 305 311	Intermittent Proposed 306	Dry 307 313	Indefinite (Fluctuating) 302	U	U		
Man-made Areas *	7	Sewage & Filtrat. Beds Perimeter Lines	Sewage & Filtrat. Beds Interior Lines	Salt Evaporator Perimeter Lines	Interior Lines 0.008"	Cranberry Bog Perimeter Lines	Rice Paddy			
Aqueducts	6	above - gnd	Elevated	below gnd	Tunnel	U				
Flumes, Penstock, etc.	5	above - gnd	Elevated	below gnd	U					
Falls	4	Large	Small	U						
Rapids	3	Large	Small	U						
Springs	2	Spring +								

* - Indicates Areal Data

\+ - Indicates Continuous Lineal Data With Discontinuous Identification Characteristics

U - Unassigned Code

TABLE III. SAMPLE CLASSIFICATION FOR DRAINAGE

JOG #	SUBCLASS-TYPE
701	9-1
702	9-2
703	9-3
704	9-4
705	9-5
706	9-6
707	9-7
708	9-8
709	5-1
710	5-2
711	Symbol
712	Symbol
713	Symbol
714	5-3
715	Symbol
716	Symbol
717	5-4
718	Roads 7-1
719	Roads 7-1
720	Roads 7-1
721	Roads 7-2
722	Roads 7-3
723	6-1
724	6-2
725	6-2
726	6-3
727	6-4
728	6-5
729	Symbol
730	3-1
731	Symbol
732	Symbol
733	3-2
734	7-1
735	7-2
736	10-4
737	10-5
738	10-6
739	10-9
740	7-3
741	7-4
742	7-5
743	7-6

JOG #	SUBCLASS
750	Symbol
751	Symbol
752	8-1
752A	4-1
753	8-2
753A	4-2
754	8-3
754A	4-3
755	8-4
756	8-6
757	Symbol
758	Symbol
759	8-7
760	8-7

CLASS - CULTURAL FEATURES

SUBCLASS	COLOR	TYPE GROUP 1	2	3	4	5	6	7	8	9
Levees Dams Piers	10	Levee	Levee w/road	Levee with railroad	Dam Lock	Dam with road	Large Dam Perimeter	U	U	Piers, Breakwaters Jetties, etc.
Railroads	9	Normal gage single track operating	Normal gage multi track operating	Narrow gage single track operating	Narrow gage multi track operating	Normal gage single track non-operating	Normal gage multi track non-operating	Narrow gage single track non-operating	Narrow gage multi track non-operating	
Boundries	8	Inter-national	Interterritorial Intercolonial Autonomous Zone	Primary Administ.	Secondary Administ.	Tertiary Administ.	Reservation	Insular		
Lineal Landmark Features	7	Pipeline above gnd	Pipeline below gnd	Powerline with pylons	Telephone or Telegraph line	Prominent Wall	Prominent Fence			
Ferries and Fords	6	Ferry single line stream	Ferry double line stream	Ferry very wide open water areas	Ford single line stream	Ford double line stream				
Miscellaneous Rail and Cable	5	Railroad siding parallel mainline	Railroads in juxta position	Car line	Aerial Cable, ski lift, conveyor-belt, mono-rail, etc.					
Boundaries in Location Diagrams	4	Inter-national	Intercolonial Interterritorial	Primary Admin.						
Miscellaneous Areas	3	Landmark Outline	Reservior non-water							

U - Unassigned Code

TABLE IV. SAMPLE CLASSIFICATION FOR CULTURE

JOG #	SUBCLASS-TYPE
[illegible]	10-2, [illegible]
[illegible]	10-3, [illegible]
[illegible]	10-4 [illegible]
[illegible]	10-5, [illegible]
[illegible]	10-6 [illegible]
[illegible]3	10-7 [illegible]
[illegible]4	10-[illegible] [illegible]
[illegible]4A	4-[illegible]
[illegible]5	5-[illegible]
[illegible]6	Symb[illegible]
[illegible]	Symb[illegible]
[illegible]	[illegible]
[illegible]	8-7
[illegible]20	[illegible]
[illegible]21	6-[illegible]
822	6-2
823	Symb[illegible]
824	Symbo[illegible]
825	Symbo[illegible]
826	Symbo[illegible]
827	Symbo[illegible]
828	Symbo[illegible]

CLASS - ROADS

SUBCLASS	COLOR	TYPE GROUP 1	2	3	4	5	6	7	8	9
Principal Roads	10	Dual Highway	All weather hard surface multilane	All weather loose surface multilane	All weather hard surface single lane	All weather loose surface single lane	Fair weather loose surface	Cart Track	Footpath	
Secondary Roads	9	As above	As above	As above	As above	As above	As above	As above		
Roads Under Construction	8	As above	As above	As above	As above	As above	As above	Classification Unknown		
Bridges, Tunnels, etc	7	Bridge or Viaduct	Underpass	Tunnel						
Roads In Outlined Developed Areas	6	Through Roads	Interior Streets							
Miscellaneous Roads	5	Unclassified								
Miscellaneous	4	Interchange perimeter								

TABLE V. SAMPLE CLASSIFICATION FOR ROADS

(2) Feature Symbolization and Annotation

The classification structure stipulated by the SOW and described above constitutes a means for identifying individual feature types. It is worthy of note that this can provide excessive identification in some cases, and incomplete identification in others, depending on the particular utilization requirements within the ACS.

Specifically, if identification is required only to accomplish output lineweight symbolization, then complete type identification is moderately excessive; definition to the type group level, as shown in Tables II - V, is adequate. Conversely, if full output annotation - such as labelled contour values - is required, then identification beyond the proposed code structure is required.

In short, whereas the proposed code structure provides a broad and useful expansion of the current machine identification capabilities, and adequately meets the needs of the Line Finishing Subsystem, it does not have the capacity for including information (primarily alpha-numeric annotation) beyond the type level. Such information would still have to be added to the digital record by semi-automatic means, and be symbolized for output by such means as the Type Placement Subsystem.

(3) Feature Representation

Since the chosen color code must somehow be physically associated with specific cartographic data, it was necessary to consider the manner by which such data are conventionally represented at the base manuscript level. Although cartographic features are categorized in terms of point, lineal, and areal data, those terms do not necessarily convey the actual form of the data as it appears on the base manuscript. That is, point data are normally represented by a line intersection, such as a small cross, while areal data are shown in the form of a bounding perimeter line, with annotative information to signify on which side of the line lies the area.

It is seen then that the vast majority of features are represented primarily by line form data, with some instances having additional information represented in the form of line intersections (points) or to-the-left or to-the-right line polarity (areas). It was therefore concluded that the problem of code association be addressed primarily in relation to line form data. At the system design level, this meant that the most favorable point for code-to-feature association would be during the linealization phase.

b. Code Representation

Having generally established the code structure boundaries, the next area for consideration was the manner of code representation. The investigation of alternate code representations had to consider both encoding and decoding requirements, and resolve the conflicts among the human factors, hardware, and software

areas of influence.

One of the key problem areas was the requirement for multidimensionality. That is, the code had to be manually composed and applied for the encoding process, and both manual and machine-interpretable for the decoding process. The tradeoffs involved in resolving this problem area are set forth below.

(1) Human Factors

When supplanting the present procedure with a color code scheme for automatic feature identification, the primary human factors objective was to introduce as little change as possible, and not increase the manuscript preparation time. A reduction in preparation time would of course be desireable, but is not the paramount concern.

The present approach to providing base manuscripts with the appropriate cartographic identifying information is for the cartographer to annotate the specific features in longhand. The tendency is to use the margin or open areas for notes and draw arrows to the feature to which the comments relate.

As such, the identification function is seen to be a twofold operation. First, it has to represent the code structure defined above by some sort of code symbology. Second, it has to provide the means for associating the actual code symbol with its corresponding feature symbol. The current manual procedure represents the information primarily by code shape, i.e., alphanumerics and cartographic symbols, while the association is accomplished by the connecting arrow. In defining the human factors problem areas, these two operations can for the most part be addressed separately.

(a) Code Definition

Since the recognition of handdrawn alphanumerics, as employed in the manual operation, is not readily implementable, this was one area where significant change appeared mandatory at the outset. The permissable avenues for change were determined on the basis of human factor capabilities and limitations. Primarily, these are:

Capabilities

- Color discrimination commensurate with the hardware
- Shape interpretation superior to state-of-the-art hardware or software.

Limitations

- Positioning capability less than the hardware.
- Size determination less than the hardware.
- Repeatability of shape, position, and size less than the hardware.

The conclusion was therefore to exploit color discrimination to the maximum, since this capability was also shared by the hardware. Of equal importance was the decision to make the code representation as independent as possible of complex symbol construction and stringent placement, since these requirements would severely tax the limits of the human element of the operation.

(b) Code Association

The most logical method of code association is by physical proximity of the code and subject feature. The principal reason this is not used more in the manual process is because of the clutter that would result. Preliminary analysis of sample charts showed, however, that with any reasonably efficient and straightforward code structure, the code symbology should be compact enough to place on the manuscript area along with the feature data. In other words, if the objective of code compactness could be met, then the preferred technique of association by proximity or superposition, of code and feature data could also be utilized.

(2) Hardware

The ACSD has been in operation at RADC for over two years, during which time considerable effort has been placed on assessing its performance. It is therefore believed that its capabilities and limitations have been determined to a high degree of certainty. These are listed below in the same organizational groupings as used for human factors in order to facilitate their comparison.

(a) Code Definition

Capabilities

- Color discrimination of ten (10) colors plus black, against a white background. The list of color types currently preferre for best ACSD operation are given in Table VI below.

TABLE VI

CURRENTLY PREFERRED LIST OF COLORS

COLOR	PENCIL TYPE	
Black	Verithin	747
Orange	Verithin	737
Red	MARS lead	1921
Magenta	Verithin	759
Lilac	MARS lead	1927
Lavendar	Verithin	742 1/2
Violet	Verithin	742
Blue Violet	Verithin	760
Azure Blue	Verithin	741 1/2
True Blue	Verithin	751
Canary Yellow	Verithin	735

- Position and lineweight accuracy and repeatability superior to the human element responsible for positioning the manuscript data.

Limitations

- Spurious data inclusion due to pencil fleck "noise"
- Spurious data dropout due to marking color irregularities
- Color mixing wherever dissimilar colors appear adjacent to one another; characterized by dropout of the intended color elements and generation of a spurious cross-breed color element

(b) Code Association

Capabilities

- Sufficient accuracy to ensure code-to-feature association by positional adjacency.

Limitations

- No absolute positioning capability as manufactured. Prohibits ready alignment of feature and/or code data originating from different manuscript scans. Correctable by field change, if required.

As previously stated, the most favorable method of code association is by proximity association. Furthermore, the most fool-proof method of proximity association is by actual superposition of code and feature data. As noted in (a) above, however, the superposition of dissimilar colors on the same manuscript, and scanned at the same time, causes color mixing and could result in erroneous color codes. A desireable alternate solution which permitted superposition, but eliminated the color mixing, was to prepare feature data and code data on separate manuscripts and perform separate scannings. This approach requires reasonably accurate machine registration between the separate manuscripts, and as noted in (b) above, this capability was not initially available. Because of its obvious desirability, however, the necessary machine retrofit was proposed to RADC and accomplished in time for application to the final color code implementation and demonstration.

(3) Software

The raster-to-lineal conversion software which restructures the raw ACSD data to the MMS format required for ACS use was initially implemented under the Computer Assisted Scanning Techniques (CAST) program. This is essentially prototype software which was designed with a relatively straightforward line connection algorithm, but incorporated considerable parametric variability to permit its utilization with - at that time - unknown machine performance characteristics. These machine characteristics have now largely been ascertained, the principal anomalies being as listed under (2) above. The software characteristics, as judged in relation both to these known machine characteristics and the code definition and association requirements, are described below.

(a) Code Definition

Capabilities

- Full utilization of previously described hardware color discrimination capabilities.

Limitations

- Spurious color flecks or color mixing could cause erroneous codes to be read
- Minimum practicality for recognizing code shape

(b) Code Association

Capabilities

- Currently associates and linealizes independent raster elements on the basis of their positional "neighborhood". Same basic logic can be applied to the code-to-feature association requirement.

Limitations

- Considerable line segmentation during the linealization process, arising from both hardware and software causes. Because this disrupts the association of all segments of a line with one another, it would likewise disrupt the association of a color code to other than the closest line segment of a particular line.
- Direction of linealization by the software is largely indeterminate, therefore nullifying significance of "--- on the left/right" type of coding information currently employed.

The fact that there was minimum practicality in attempting to recognize code shape, and that a compact code was necessary to facilitate proximity association, led to the preliminary conclusion that a simple color "slash" or "blob" would be the most functional code appearance.

Also, the segmentation which occurs during linealization pointed to a potential problem area - namely, that of perpeturating a locally applied code along the full length of any lineal feature. In anticipation that redundant coding along the line might be the most convenient way to compensate for this segmentation effect, the feasibility of such redundant coding was slated as an item for consideration during the code design and implementation phase.

3. PROBLEM SOLUTION

a. Specific Identification Requirement

As already established, the basic identification requirement was that "A way of color coding the feature data must be developed which allows recognition and identification of the feature types." For the feature types described by the JOG Series 1501-A specifications, the following kinds of identification required consideration.

1. Cartographic Type Identifier

The JOG series 1501-A charts identify approximately 250 types of feature data and give their output symbolization. However, this may not equate directly to the level and/or manner of identification desired of a coding scheme. The following alternatives were considered.

(a) Encode all 250 JOG series 1501-A feature types -- permits retention of full cartographic identification. For expandability and code assignment flexibility, the code structure should be designed to accommodate from 300-500 feature types.

(b) Encode feature types uniquely by class and commonly by types within a class having the same output symbolization -- this reduces the number of identifiable types to approximately 180, yet permits all identification required for output symbolization, which is the requirement stipulated by paragraph 4.2 of the SOW. Table II-V are examples of a hierarchial structure based on such a type grouping.

✱ Conclusion

The code structure should be designed to accommodate (a); implementation of (b) is more appropriately (and flexibly) accomplished via software tables.

2. Geometric Type Identifier

Identification is ultimately required for all point, lineal, and areal features. Of these, there is an obvious similarity at compilation between lineal and areal features, the latter being represented by its boundary line plus some indication as to which side of that line lies the areal information. The results of the coding technique investigation was that a common coding scheme appeared equally applicable to both lineal and areal data. This was not the case with point data, which tended not to fit the coding schemes which appeared most desirable for both lineal and areal data.

* Conclusion

Limit the coding requirements to lineal and areal data, which constitute the bulk of the data identification problem. Refer point data entry and identification to an earlier point in the chart compilation (e.g., a keyboard, rather than a plotted coordinate entry). Since there are approximately 210 lineal/areal feature types, the coding structure should still be in the range of 300 - 500 type classification.

b. Primary Identification Alternatives

A broad review of the basic feature identification requirement, made with an awareness of existing and potential ACS capabilities, showed that there existed two primary alternatives for satisfying this requirement with data entered via an ACSD-type input device. It was presumed that these alternatives had been investigated and evaluated prior to the issuance of this contract, since it is their resolution which suggests a color code approach. These alternatives were formally restated however, to permit the suggestion that the rejected alternative could still be effectively utilized to "back up" a color code scheme where that scheme either unintentionally fails or is awkward to implement.

(1) Pre-input Tagging

This was the inferred approach; it entails some manner (color, shape, etc.) of graphically encoding the base manuscript feature data, where the coding may be done either on the manuscript itself, or an overlay to that manuscript. This approach for the most part presumes that the color detecting capability of the ACSD can be further exploited -- and efficiently so -- to satisfy nearly all identification requirements. Another significant factor, and perhaps the decisive one, is that such a color-coded graphical identification scheme would be most compatible with the color-coded analog data file which is being investigated for future ACS implementation.

(2) Post-input Tagging

This was the discarded approach. The concept was to utilize the ACSD/CAST input system primarily for bulk entry of partially identified feature data, and complete the identification after linealization using a Cartographic Digitizing Plotter (CDP) type verify/edit device. Although the present operational software for the CDP makes it grossly ill-suited for such an application, the proper re-programming of that software could make this solution quite feasible. The general approach would be to:

(a) Serially accept the unordered data strings output by CAST. Fully concatenated feature segments would be preferable, but are not necessary.

(b) Slew the CDP crosshiar to the start point of each feature and wait.

(c) The operator/compiler visually identifies the feature which appears under the crosshair, on the actual base manuscript, which has been positioned on the CDP and registered to the required coordinates. If the start point is ambiguous, as at an intersection, he may advance the crosshair pointer a given number of data points to permit unambiguous identification.

(d) He then keys in the required identification code using the gantry-mounted keyboard or teletypwriter. Properly implemented, this procedure should be no more time consuming than the application of graphical color tags to the base manuscript.

(e) Slew to the start point of the next serially accepted data string and repeat.

The major objections to this approach are: (1) it does not fully exploit the ACSD color capability, (2) it requires extensive redesign of the CDP operating system, and (3) it is not compatible with a graphically oriented archive. Taken together, these objections discourage the use of post-input tagging as the primary approach.

✱ Conclusion

Implement the graphically color-coded, pre-input tagging scheme described herein, but also consider the CDP approach for verify/edit and general backup to the primary approach.

c. Color Code Strategy

The general strategy taken for devising a technique for pre-input tagging was to build upon, or better utilize existing capabilities while avoiding, or correcting for existing limitations. At the same time, consideration had to be given to the procedural guidelines and classification hierarchy which had been previously investigated and are documented under Problem Definition. The investigation of techniques for accomplishing pre-input tagging suggested a major division into just two general approaches: in-line versus associative coding. The more important areas for investigation and evaluation proved to be the individual techniques by which these general approaches might be implemented.

(1) In-line Coding

This technique required that all base manuscript lineal data be serially color coded at compilation time. Serial color coding means that the compiler

would draw most of a given feature using a given first color, but at some point along the line switch to a second color, then a third color, etc., and finally back to the first color. The number of colors required would depend on the structure of the hierarchy and whether an ordered or unordered code structure were used. (These considerations are identical for in-line and associate coding, and are considered in greater depth when discussing associative coding). This approach respresented the most direct extension of the present hardware/software capability; a modified approach to line connection could be used to serially link the color coded section and combine or permute the individual color segments in order to provide the required number of identifications. The use of a single manuscript would permit identification with the minimum scan time. The major disadvantage is seen to be in the revised and probably lengthier compilation procedures which this approach entails.

(2) Associative Coding

This approach utilizes separate "feature" and "symbol" information. Feature compilation entails essentially the same procedures which are used in the manual process. Symbol annotation entails placing the required symbol code on the manuscript, or an overlay to the manuscript, so as to permit association of the symbol with the required feature. The following areas required consideration.

(a) Symbol Characteristics

- Color -- should utilize existing ACSD color detecting capability to its maximum extent. The possibilities are:

  - Basic 10 colors -- use 10 colors for code combinations, reserve black for possible future editing. (Black is most applicable because it does not color mix).

  - Basic 10 plus black -- if reserving of black for editing is not desired or, in case of overlay, black may be used on overlay and reserved on manuscript.

  - Basic 10 plus blank -- acknowledge blank on overlay only. Intent would be to assign the blank designations to high usage features, thereby reducing coding required during compilation.

- Shape -- should be easy to hand generate and/or position during compilation, and also easy to recognize at processing time. Latter confirmed the

undesirability of attempting any type of pattern recognition; the former suggested a straightforward dot or line code.

- Position -- should be incorporated as necessary to:

  - Associate symbol with appropriate feature. Association of a symbol to a feature would be on the basis of physical adjacency. Adjacency tolerance should be broad enough to permit symbol positioning errors and possible misregistration (overlay technique only), yet tight enough to ensure correct association. Symbol positioning tolerance of ± 1/16 inch should be adequate.

  - Associate separate symbol components with one another. If each component can be symbol-to-feature associated, then symbol-to-symbol association is implicit; however, the order of such association is not. The alternatives were to design for (1) unordered combinations only, or (2) ordered combinations, where the recognition of symbol order is part of the design requirement.

(b) Symbol/Feature Differentiation

If an overlay manuscript were used, distinguishing between symbol and feature data could be readily accomplished by separately scanning the base manuscript and its coded overlay. If a single manuscript were used, the symbols must be made physically distinguishable from the feature data. The two most probable ways to accomplish this would be to:

- Allocate separate colors for feature and symbol data -- this facilitates identification but drastically limits the combinatorial capabilities of the code, as shown in the examples in paragraph d of this section. It was determined to be an inadvisable approach.

- Identify symbols by size -- by specifying the size of the color symbols to be larger than "noise" flecks, but smaller than legitimate linear features, software tests could be devised to recognize symbol data on the basis of upper and lower size limits. This seemed to be a more desirable approach than that described above; still it was sanctioned with caution because it

further complicated the already difficult problem of differentiating between valid feature data and noise data.

(c) Single Manuscript Versus Coded Overlay

In addition to symbol/feature differentiation, there were several other factors involved in the trade-off between a single manuscript and a coded overlay approach. Figure 1 portrays graphically the factors comprising the trade-off and resolves it in favor of the overlay approach. It can be seen that a single, coded, base manuscript has the obvious advantages of minimum scan time and no registration requirements prior to feature/symbol association. A coded overlay takes about twice the scan time and requires some form of registration for symbol/feature correlation; however, it has the significant advantages of:

- Inherent symbol/feature differentiation -- which was shown in the preceding paragraph to present a problem in the case of a coded base manuscript.

- Straightforward symbol/feature association -- the association of a symbol to its intended feature can be made on the basis of intersection (i.e., overlay superposition) rather than by proximity. For randomly oriented lineal features, this significantly eases the software association logic and operationally should improve the likelihood for successful association.

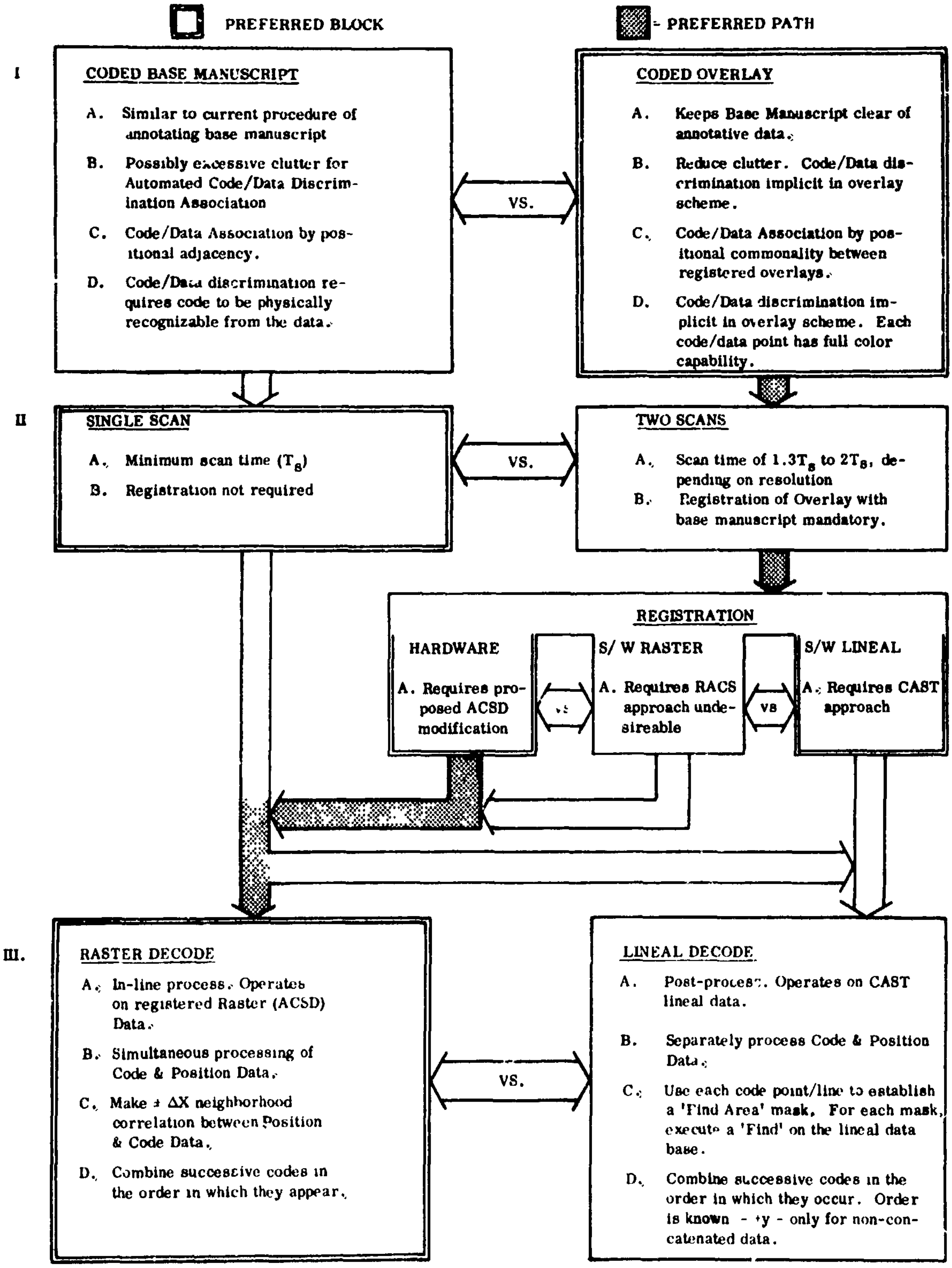
✱ Conclusion

The above considerations resulted in adopting a color code strategy which would use associative, rather than in-line, coding and utilize a coded overlay to the base manuscript, rather than coding directly on that manuscript. As for the symbol characteristics, color alone would be utilized to convey identifying information. Shape would remain largely undefined, to be determined by individual user preference. Size and position would be as required to ensure superposition of the feature on the base manuscript and the symbol on the overlay, taking into account possible registration inaccuracies. In order to properly implement this strategy, the necessity for machine registration was recognized, and subsequently fulfilled by adding a pin registration capability to the ACSD under separate procurement.

FIGURE 1

CONSIDERATIONS FOR

CODED BASE MANUSCRIPT VS. CODED OVERLAY



d. Combinatorial Possibilities of the Color Code Strategy

It was seen that strategy might be implemented using any of several combinatorial approaches. The several levels of the code might be considered in an ordered ($AB \neq BA$) or unordered ($AB = BA$) fashion, and might or might not permit replacement (i. e., color repeats in the code). The relationships from basic combinatorial theory are:

Ordered --

With replacement: $P_r^n = n^r$

Without replacement: $P_r^n = (n)_r = \dfrac{n!}{(n - r)!}$

Unordered --

With replacement: classically undefined

Without replacement: $C_r^n = \dfrac{(n)r}{r!}$

where n is the total number of colors and r is the number of levels of the code. The following trade-offs between ordered and unordered, with replacement and without replacement, were considered.

(1) Ordered versus unordered -- ordered arrangements possess r ! times as many combinations as unordered arrangements; however, an ordered situation is significantly more difficult to implement. At input, order would have to be interpreted as either order of appearance in the raw raster data, or order of appearance along a scanned line segment. The former places close tolerances on scan orientation and symbol placement; the latter entails line direction, which is meaningless to the compiler and difficult to predict without careful inspection.

✱ Conclusion -- structure the code and hierarchial arrangement to operate with unordered combinations, if at all possible.

(2) Replacement versus no replacement -- code arrangements with replacement produce more combinations than without replacement, but there is not so drastic a difference as in the case of ordered versus unordered data. From the standpoint of implementation, the no replacement case has two advantages. First, it is easier to implement: each color detected can be logically or'ed with all others without concern as to whether it represents a new symbol or a continuation of a previous symbol. Second, it permits the hierarchy to be modified for a limited degree of

variable level coding if desired. In essence, this acknowledges the absence of a color code, and interprets it as a specific code assignment. For example, whereas green-red-blue would have a specific code assignment, likewise would green-red-blank and green-blank-blank. These abreviated codes could be allocated to frequently appearing features and thereby reduce symbol compilation time.

* Conclusion -- structure the code and hierarchial arrangement to operate without color repeats. Include the variable length code as an optional feature.

e. Synthesis of Required Code Structure

Table VII shows the unordered combinations, with no color repeats, which can be obtained from a two-level symbol code employing ten colors (45 combinations), ten colors plus black (55), and ten colors plus black plus blank (67).

Color 1 - Feature

Color 2 - Symbol 1

Color 3 - Symbol 2	1	2	3	4	5	6	7	8	9	10	Black	Blank
1												
2	1											
3	2	3										
4	4	5	6									
5	7	8	9	10								
6	11	12	13	14	15							
7	16	17	18	19	20	21						
8	22	23	24	25	26	27	28					
9	29	30	31	32	33	34	35	36				
10	37	38	39	40	41	42	43	44	<u>45</u>			
Black	46	47	48	49	50	51	52	53	54	<u>55</u>		
Blank	56	57	58	59	60	61	62	63	64	65	66	<u>67</u>

TABLE VII

COMBINATIONS FOR TWO LEVEL SYMBOL CODE

Obviously, some additional level of coding is needed to achieve the required 300 - 500 combinations. Two cases were considered: one where all coding is provided by symbol coding, and the other where the additional coding is accomplished by feature coding.

(1) Symbol Coding Only

Presuming that it might be desirable not to code the base manucript feature data, consider the combinatorial implications and number the code levels required to implement an unordered code structure using symbol coding only. The number of combinations C^n_r are shown below for relevent values of n and r.

C^n_r		r				
		2	3	4	5	6
	10	45	120	210	252	210
n	11	55	165	330		
	12	66	220	495		

It can be seen that the ten plus black, and ten plus black plus blank coding schemes are implementable using a four-level-code, but that a ten color scheme is not implementable at all using straight combinations.

(2) Feature/Symbol Coding

The necessity for feature/symbol differentiation has been previously established. This fact pays a combinatorial dividend in that the unordered symbol combinations may be combined with a feature code in an ordered fashion. Consequentialy, with this approach, the total number of available combinations is equal to the product of the selected number of symbol combinations (45, 55, or 67) and the number of colors used for feature coding (10 or 11).

✱ Conclusion -- Use the unordered, no color repeat symbol code (45 combinations, plus 12 optional slots for variable length code assignments) with a 10-color feature code. This provides 450 code assignments within a three-level hierarchy, without utilizing black.

f. Code Hierarchy Assignments

As previously established, "To the extent possible, the color tags used shall have a hierarchial structure. That is, the appearance of a particular color, e. g., blue, or as the first color in a sequence should indicate that a particular class of feature, e. g., drainage, follows." For the JOG Series 1501-A charts, the breakdown

of the overall number of feature types by class is as shown in Table VIII.

Table VIII

JOG Series 1501-A Feature Type Breakdown

Feature Class	Number of Feature Types	
	Total	Lineal + Areal Only
1. Drainage	50	48
2 Relief	57	49
3. Culture	62	51
4. Roads	29	25
5. Coastal Hydrography	27	17
6. Aeronautical	17	10
Total	242	200

Note that the first three classes have too many feature types to permit the prescribed two-level symbol coding. However, if each of these classes is split into two major subclasses, the breakdown can be reorganized as shown in Table IX.

Table IX

JOG Series 1501-A Feature Reclassification

Feature Class	Number of Feature Types	
	Total	Lineal + Areal Only
1. Drainage A	25	24
2. Drainage B	25	24
3. Relief A	29	25
4. Relief B	28	24
5. Culture A	31	26
6. Culture B	31	25
7. Roads	29	25
8. Coastal Hydrography	27	17
9. Aeronautical	17	10
Total	242	200

This breakdown is still reasonably consistent with the stated hierarchial objective, and has the advantage of a more even distribution of feature types among the nine specified feature classes and major subclasses. Only nine of the ten available colors are assigned to a class, and of these none requires more than 60% utilization for the currently assigned feature types.

g. Supplemental Identification Considerations

The preceding paragraphs have defined a coding scheme to unambiguously identify all JOG features which are represented on the base manuscript by line form data. While this by itself satisfies the major part of the original identification requirement, there remain a few items whose further identification should also be considered. These items and their individual identification requirements are discussed below.

(1) Identification Discontinuities within Lineal Feature Data

The term lineal discontinuity may be applied to a continuously represented line on the base manuscript which undergoes a change in designation at some specific point on the line. Examples of this are roads at bridges, tunnels, underpasses, et al. Currently, these types of discontinuities are compiled in a number of different ways: tic marks, change from solid to broken lines, separate annotation, and various combinations of all of the above.

(a) Problem

The several ways of representing lineal discontinuities at compilation are not amenable to an automated procedure of recognition and tagging. A more suitable scheme, which would be compatible with the basic color-code and linealization software, was required.

(b) Approach

Previously described color-code data tags contain identification significance only, and therefore logically belong on the overlay manuscript. Since lineal discontinuity identifiers contain positional significance, it was concluded that they more logically belonged on the base manuscript with all the other positionally-related data.

The function of the discontinuity identifier on the base manuscript is to interrupt the linealization of the lineal feature in question. Based on experience with CAST software, this may be accomplished in either of the following ways:

- Denote each point of discontinuity (i.e., both ends of a tunnel) with a black line crossing the feature line.
- Draw the duration of the discontinuity (i.e., the entire length of a tunnel) in some color different from the main feature color.

Either of these techniques causes an interrupt in normal CAST linealization, although in the former approach, care must be taken to make the black line wide enough to fault the CAST lookahead.

In both cases, the resultant line segment representing the dissimilar portion of the line must be given its specific identification significance by a color code on the overlay.

(c) Implementation

Since neither of the above approaches required additional effort to implement, it was determined that both approaches should be tried in test cases to establish the preferred way, if any.

(2) Identification Discontinuities at Line Intersections

The term branching discontinuity may be applied to those cases where like colored features join or cross. Examples of this are drainage intersections, road crossings, et al. Currently, the cartographer resolves the path of a line through an intersection either by assessing the overall picture, or by the aid of supplemental notation placed on the base manuscript.

(a) Problem

The current linealization software accomplishes line connection on a single pass, in-line process by which neighboring data elements are associated. When a branching situation occurs, the software does not "know which way to go", and has therefore been designed to effect logical termination of all lines joining at an intersection.

Although the stated requirement is to "identify" the path of a line through an intersection, the real need for this capability in the automated system warrants further examination. The reason for this is that full cartographic identification can still be accomplished, in spite of line segmentation, simply by redundantly applying the color code to each line segment.

The question which then arises is whether such segmented data is acceptable to the ACS. In the original context of the Raster Processing Subsystem, where an input color code would be used primarily to determine a Raster Plotter lineweight designation, segmentation does not affect performance in the least. In the revised and broader context, where X, Y line plotting may be required, segmentation does degrade performance by introducing excessive and time consuming pen up-slew-pen down command sequence.

* Conclusion -- It was concluded that the concern with branching discontinuities lay more with the structural organization of the data than with its

cartographic identity, and as such, required a solution somewhat beyond the bounds of the color tag solution previously posed. Two potential solutions to the structuring problem were seen and are presented below.

(b) Potential Solutions

(1) Lineal pre-processor

If the segmented data only degrades in use with line plotting devices, a lineal device pre-processor could be incorporated to reduce segmentation and minimize overall slew time.

Two types of information are available to facilitate the desegmentation process: the cartographic type identifier, and the terminal X, Y points of the lineal data set. The data could be first sorted by common type identifier, and then ordered by common terminal points. Finally, the data sets would be reordered and merged to obtain a continuous locus structure. The resultant data could be output at this point, or further structured for minimum slew time.

Only one type of information is available for minimizing slew time: the terminal X, Y points. A reasonable approach would be to begin with any lineal data set, and then select each succeeding data set by its minimum distance from the preceeding data set. This should sufficiently reduce the plotter slew time; any attempt at actual minimization would probable defeat its purpose by utilizing excessive processing time.

(2) Modified Coding Procedure

The color coding strategy previously outlined utilizes one level of coding on the base manuscript and two levels on the overlay manuscript, the full three levels of code being required to represent all feature classes and types of the JOG 1501-A series.

The problem of branching occurs when two identical feature classes - as represented by the same base manuscript color - intersect. Branching, then, refers to the inability to automatically cope with monocolored intersections.

This raises a very fundamental question: should a system predicated on color-coded identification, association, and discrimination of data be expanded beyond that realm, i. e., into the area of monocolored operations? Need it be? Or is it possible to restructure the problem to permit a fully color-coded solution?

Based on discussions with RADC and ACIC technical personnel, it was concluded that such a restructuring would be acceptable, and might even be preferable. The revised structure would probably require that a separate base manuscript be drafted for each major feature class (e. g., drainage, roads, culture). In

terms of the proposed color-coded identification scheme, the implication was that the 45 type classifications afforded by the overlay code would be adequate for full feature type identification within a particular class. This would leave open the entire color-coding capability of the base manuscript for such other purposes as color identifying all line intersection. It was therefore suggested that:

- The two-level overlay color-code be utilized for cartographic feature type identification.

- The single-level base manuscript color-code be utilized, as necessary, for lineal continuity identification.

This approach has the effect of solving the mono-colored branching problem by eliminating its occurance. This is more efficacious than the lineal pre-processor approach since it provides the compiler with a level of identification beyond the prescribed type designation. In reality, identical road types can cross, and identical drainage types can join. The decision as to which branch terminates and which continues is best made by the cartographer and — by this judgement — best portrayed on the base manuscript by the above adaptation of the prescribed color code.

(3) Areal Boundary Identification

Within the ACS digital data base, areal data are defined by the perimeter line bounding that area, plus additional identifying information signifying to which side of the bounding line the areal data lies. In the Lineal Data Processing System, this information is conventionally expressed as "data on the left/right"; this is made possible by the fact that the direction of lineal trace is known at the time the areal information is entered.

(a) Problem

The term "direction of trace" is not particularly relevant to the cartographer who must color code an overlay to a base manuscript which is to be ACSD-scanned, not traced. That is, he does not immediately know, and should not be required to spend additional time determining, the eventual direction of linealization which the raster-to-lineal conversion software will take. Consequently, it was necessary to establish some means of denoting areal positions which were more amenable to the cartographer's frame of reference.

(b) Approach

Representation of Areal Data -- Areal data may alternatively be represented in terms of a single, closed, bounding line, and supplemental information indicating whether the areal data is inside or outside that line. Since the representation of an area as

on the inside or outside of a bounding line bears a greater relationship to the physical appearance of the feature, it should be a more convenient identifier for the cartographer to use when assigning color codes.

A review of the specific areal features listed in the JOG 1501-A specification confirms that this representation accurately applies to all actual cases requiring solution. The only apparent exception to this rule is where an areal feature is interrupted by a chart boundary. In that case, the chart boundary may itself be construed to be a segment of the bounding line.

• Impact on the Coding Structure

The initial investigation and evaluation established a hierarchial code structure based on the individual designations set forth in the JOG 1501 series specification. In order to represent the "area on the inside/outside" information stipulated above, the number of areal data code assignments might have to be doubled, depending on the method of implementation. The magnitude of these expanded code assignments are as shown below. The recommended hierarchy remains valid, in spite of the expanded number of codes required.

Table X
JOG Series 1501-A Expanded Feature Classification

Feature Class	Number of Feature Types	
	Lineal & Areal	Lineal + (2xAreal)
1. Drainage A	24	36
2. Drainage B	24	36
3. Relief A	25	35
4. Relief B	24	35
5. Culture A	26	32
6. Culture B	25	31
7. Roads	25	25
8. Coastal Hydrography	17	26
9. Aeronautical	10	12
Total	200	268

(c) Implementation

It has been stated that bounding perimeter lines and "area on the inside/outside" designation unequivocally describe areal data. To test this statement, consider (1) a procedure which might be used to so describe areal data, and (2) a corresponding procedure which might be used to reconstruct the physical data solely from the header information and closed lineal boundaries. The procedures given below provide working examples. Minor variations are possible depending on preference of implementation.

• Areal Data Coding

Figure 2 shows the manner in which several examples of shoreline would be coded. For illustration, only a single level code is shown. Note that for areal data such as shoreline, a code assignment is required for both inside and outside cases. For perennial lakes/ponds, however, only a single code is required, since the area is by definition on the inside. It is only necessary that the appropriate symbology be placed one time on the overlay for any closed feature. Contingent on the line connect capability to concatenate all feature segments and close the feature, that symbology will automatically be associated with the entire perimeter line.

As shown in the figure, this approach dictates that closure be drawn in by the compiler wherever an area intersects a chart boundary. For the current system, the area boundary would have to be drawn exactly coincident with the chart boundary. In a production system, it would be more adviseable to provide some overlap (as shown) and have software trim the excess at plot time.

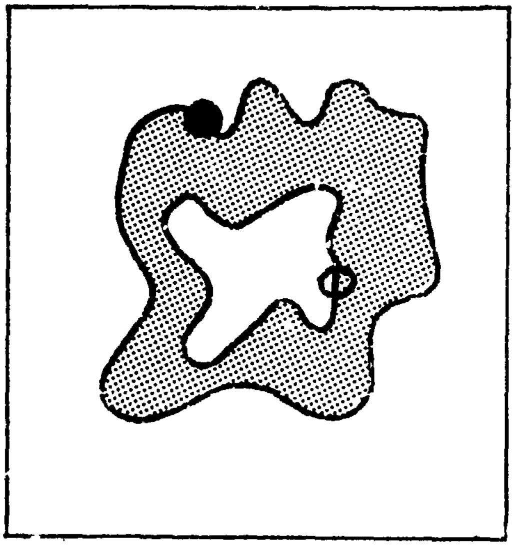
• Areal Data Decoding

The ultimate test of the areal data coding approach is the ease and completeness with which it can be decoded to physically create the intended areal data using appropriate plotting devices. Given the fully concatenated perimeter data and inside/outside header information, the following paragraphs describe the basic logic for such decoding. Although the plotting of areal data would seem practical only on raster devices, lineal devices are also discussed.

•• Raster Decoding

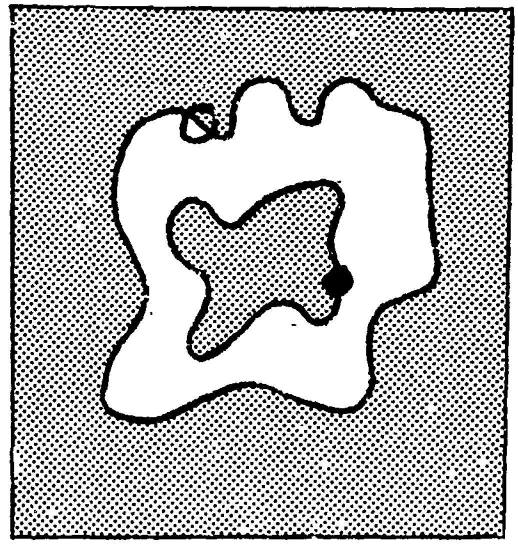
It is necessary to presume only that the raster device to be used makes a complete scan from left to right extremities, starting and ending each scan with its imaging beam off. The data provides all other information.

For an "area on the inside" feature, the beam must be switched from off to on at the point where that feature is first encountered on a given scan line, and toggled for every encounter of that feature thereafter on that scan line,



(a)

Island Within Lake

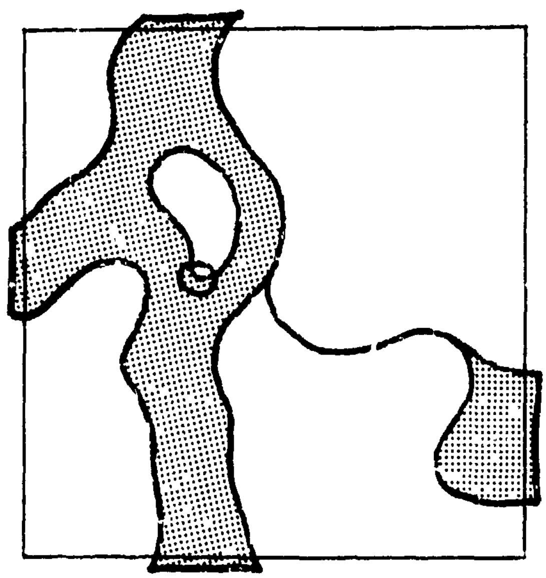


(b)

Lake Within Island

Encoded Area Outside ○

Encoded Area Inside ●



(c)

Complex Drainage

AREAL DATA CODING

FIGURE 2

including left and right extremities. Outside the feature extremities, the scan line would begin and end with a beam-off condition.

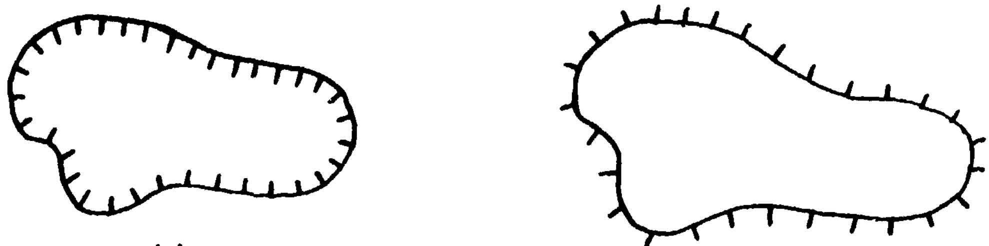
For an "area on the outside" feature, the beam must be switched from on to off at the point where that feature is first encountered on a given scan line, and toggled for every encounter of that feature thereafter on that scan line, including left and right extremities. Outside the feature extremities, the scan line would begin and end with a beam-on condition.

Any physically realizable combinations of the above type features may be handled by little more than a logical OR'ing of beam commands. The procedure would be to examine the requirement of the leftmost feature point on a scan line to determine whether an "off" or "on" startup condition is required. Once that condition is determined, each areal boundary encountered merely toggles the beam to give the correct sequence of exposed and unexposed areas. Of course, this presumes that each area had been correctly coded and fully linealized into a closed perimeter line. A more prudent method of implementation would be to maintain sufficient program bookkeeping to be able to check the desired beam transition at a point against the initial beam condition at that point, and tag any discrepancies for correction prior to actual plotting.

·· Lineal Decoding

As previously stated, true areal plotting seems applicable only to raster devices. Nevertheless, since lineal devices are normally relied upon for most editing, and since areal data may require editing, it would be advantageous if provisions existed for lineal devices to somehow represent areal data. This could be done symbolically, much in the way such data are currently portrayed in compilation, without actually "filling in" the area.

It can be seen that the inside/outside designation which readily converted to beam on/off for raster plotting is not applicable to this case. The lineal device requires on-the-left/right information, or equivalent. This being the case, the concatenation algorithm installed in CAST Engineering Change "A" was specifically designed to always linealize in a predictable, counterclockwise direction for closed lineal features. As a result, areal data coded to be "inside" directly converts to "on the left", while "outside" converts to "on the right". Special edit symbolization might then be employed to give the following representations for lineal plots of areal data.



(a) Area on the Inside

(b) Area on the Outside

Possible Edit Area Symbolization

FIGURE 3

It is worthy of note that this suggested edit symbolization bears a marked similarity to some output lineal symbolization which will be required in any event. Reference is made to such feature types as JOG 1501-A series types 416, 417, and 421, 422.

h. Utilization Requirements

An important factor in selecting a color code strategy was its human factors impact. Of prime consideration was the ease of code utilization; that is, nominal constraint was to be placed on the compiler insofar as the accuracy and procedure required in applying the code symbology.

The following set of rules completely defines the constraints which must be met in applying the prescribed color code to the overlay manuscript. There are no constraints on base manuscript preparation beyond those currently required for ACSD utilization.

(1) Overlay color patches must (when overlayed) coincide with their associated features on the base manuscript.

(2) Overlay color patches must (when overlayed) be separated by at least two (2) linewidths from all non-associated features on the base manuscript, and must also be separated a like amount from one another.

(3) A maximum of two (2) patches of different colors may be associated with a single base manuscript feature.

(4) Redundant coding (repeats of the same color(s)) is permissable and should be practiced as necessary to ensure identification of all feature segments.

(5) Color patches may be applied in any size, shape, color, and frequency so long as they do not violate any of the preceding rules.

4. ASSESSMENT OF THE SOLUTION

The color coding study and investigation has postulated a coding scheme which would utilize one level of coding (10 colors) on the base manuscript, and up to two levels of coding (10 colors each, but no color repeats within any two-color code combination) on a mechanically registered overlay to the base manuscript. This provides forty-five (45) identifiable combinations for the overlay code by itself, and 450 combinations when taken in conjunction with the base manuscript code. The ten-color coding scheme presumes no specific color assignments; these are to be provided according to the preference of the contracting agency. It was suggested that the "color" black be witheld from use in the coding scheme. This recommendation was made on the basis that black alone does not color-mix with other colors, and hence has potential for utilization as a special "edit color" in this or future applications, if required.

The coding format described is considered to provide those characteristics required of a system to function efficiently in the ACS. The format is hierarchial in structure with three levels of feature classification which satisfactorily accommodate a natural segregation of cartographic feature types. The structure permits ready table look-up of code for any feature type, and for the identification of a coded feature on the manuscript.

The coding format has been designed to be highly versatile and flexible. Coding assignments within each first-level classification may be modified as desired to group feature types on a different basis than that described. Color may be employed at the first level of coding to provide an easy recognition of feature class; at second-level coding, color assignments may be made to provide a consistent use of color in all feature classes to relate to lineal and areal features. When specifications change, the format may be readily changed to meet new requirements. The coding format described allows all features to be shown on one compilation manuscript. For those cases in which more than one manuscript is necessary because of feature density, a separate coding format may be developed for each manuscript each format utilizing a complete structure with 10 colors.

The format presented is expandable, permitting a logical extension to coding of feature types not included in this tabulation. This capability will allow the refinement of type identifications as experience is accumulated, and ACS operating procedures are developed. The format lends itself to a logical expansion to utilize any increase in color recognition capability of the ACSD, or the development of color pencils or inks which better utilize the ACSD potential.

The code utilization is straight-forward from the standpoints of both application and interpretation. The accuracy requirements in the compiler's positioning of the code, and of the ACSD's registration of the related manuscripts, are less exacting than most other aspects of the cartographic process. This should result in a feature identification error rate which is very low by normal editing standards. In all, it is believed that the proposed scheme comprises a workable solution which attains a good balance among its constituent cartographic, human factors, hardware, and software elements.

SECTION III

FEATURE IDENTIFICATION SOFTWARE SYSTEM

1. INTRODUCTION

The Feature Identification Software System (FISS) constitutes a set of stand-alone software programs which accepts ACSD-generated raster formatted data records, extracts and correlates color tagged positional data based on the hierarchial color scheme established in Section II,and connects these data elements on a point-to-point basis to form lineal formatted data records which are symbolically tagged by feature type. The major system components required for the generation of the lineal files are described in Paragraph 2, while the Feature Identification Software System operation is described in Paragraph 3.

2. DESCRIPTION

The major system components required for the generation of the lineal formatted files which serve as input to the ACS Lineal Processing Software System are the Automatic Color Separation Device (ACSD) and the Feature Identification Software System (FISS). These functional components are described in the following paragraphs.

a. Automatic Color Separation Device

The function of the ACSD is to generate the raster formatted color files which serve as input to the FISS. For the most part, the constraints on ACSD operation are the same as required for the existing CAST linealization software. There are, however, two significant areas where the operational procedures differ. These areas address the additional requirements for absolute machine registration and certain hardware parameter settings and are discussed below.

(1) Machine Registration

The prescribed manuscript compilation procedures yield a base manuscript and coded overlay manuscript which must be separately scanned, but properly registered at processing time to obtain valid FISS results. Because software registration (rotation in particular) of point represented data is quite inefficient, a system of machine registration was adopted based on the ACIC pin registration system.

This was accomplished by outfitting the ACSD with the standard ACIC "pin" arrangement to permit fixed, repeatable manuscript mounting. Scan repeatability (i.e., registration) between separate scannings may then be obtained by the following operational setup procedures.

- Registration in X is achieved by fixing the photodetector positioning rings for the successive scans of base and overlay manuscripts.

- Registration in Y is achieved by zeroing the Y-counter and positioning the scan head at its upper limit of travel for all scannings.

(2) Hardware Parameter Settings

Certain ACSD parameter settings must be given additional consideration when preparing data files for FISS inputs.

Since the two separate data files mustbe subsequently associated, both the base and coded overlay manuscripts require scanning with the same settings for resolution and minimum area.

Because extraneous color data on the overlay manuscript could easily be interpreted as a valid color code, only those color switches representing valid overlay colors should be enabled during the scan of that manuscript. Since base manuscript colors are selectively extracted during FISS execution, all color switches can be enabled during the scan of that manuscript.

b. Feature Identification Software System

The function of the FISS is to generate the lineal formatted feature file which serves as input to the ACS Lineal Processing Software System. The input to the FISS program is properly machine-registered ACSD raster data files. FISS is a stand-alone program written for the ECF PDP-9 computer facility.

(1) Design

FISS is designed with CAST as its basis for point-to-point correlation. New software was designed and written to implement the color tag correlation and identification functions of the FISS. The input end of CAST was modified to accept two ACSD-generated raw raster data files in place of individual RACS-generated color files. This modification eliminates the need for the RACS color separation function completely, and permits the correlation of the pre-registered positional data and color tag data to be performed in an in-line fashion at the raster level.

(2) Implementation

Appendix I describes the Feature Identification Software System operating instructions. Previously established CAST line connection control parameters are selected through an operator/program query-response mechanism. In

addition, a base manuscript feature color which identifies a feature class resident on the base manuscript is requested for each serial pass of the ACSD input files. Operation occurs in three functional phases as described in Paragraph 3.

3. FISS OPERATION

The FISS program as designed is divided into three functional phases of execution: initialization, feature data set generation, and MMS tape generation. These three functional phases of execution, plus the generated statistical data, are described in the following paragraphs.

a. Initialization

The initialization phase of the FISS program provides a software controlled user-system interface to query administrative messages via the console teletype/teleprinter and to enter control information and data. The control parameters required for feature data set generation and the base color representation of a specific class of feature types are entered during the system initialization phase. Error diagnostics identify abnormal user entries and instruct user recovery capabilities.

b. Feature Data Set Generation

This next phase of the FISS program is devoted to the generation of the intermediate data set structure. During this operation, the functions performed include the input and sychronization of base manuscript and color overlay raw raster data records, point-to-point correlation of base manuscript positional data (line connection), correlation of feature identification data (color tags) to feature positional data, and transfer of this feature identification data to disk-resident feature data sets.

(1) Raster Data Input

Paragraph 2.a described the procedure for producing two ACSD raw raster data tape files in proper registration. The one tape file contains feature positional data resident on the base manuscript, while other contains feature identification data (color tags) resident on the color overlay. The two tapes are synchronously processed in a serial batch mode and supply the feature positional data input required for line connection and the feature identification data input required for color tag correlation. For each serial pass of the ACSD input files, a single base feature color is extracted, linealized and correlated to the overlay color tags.

(2) Line Connection

The CAST raster-to-lineal conversion package serves as the basis for the line connection functions of the FISS program. The point-to-point

correlation criteria is applied to the base manuscript positional data to create a disk-resident lineal data set structure which describes the base manuscript data. Only those line connection capabilities existent in CAST at the time of its induction into the FISS program are implemented in the current version of the FISS program.

(3) Color Tag Correlation

Color tag data resident on the color overlay must be correlated to the feature data sets created from the positional data resident on the base manuscript. The basis of this correlation is the positional adjacency of the color tag data to the base manuscript feature data. This positional adjacency is ascertained by the same lookahead mechanism used to line connect the individual feature data elements. A feature data set may have from zero to two unique color tags legally associated with it. More than two unique color tags correlated to a single data feature indicates erroneous color tag information.

Color tag information correlated to feature data sets is stored in a core-resident color word dedicated to that feature data set buffer. The color word reserves a bit for each of the possible color codes generated by the ACSD as shown in Figure 4 . A color tag correlation sets the appropriate color word bit for the feature data set indicated. The feature color word structure enables the accumulation of all possible color tags correlated to a particular feature data set.

(4) Mother Block Update

Feature line segments are represented as feature data set blocks (mother block and its associated daughter blocks). Upon unconditional feature data set termination,the mother block update function transfers the core-resident color tag information accumulated for that feature data set to its mother block. The color tag information overwrites a free word of the mother block header. Linkage information required by the output phase is also transfered to the mother block header. The feature data sets remain disk resident until the output phase of the FISS program is initiated.

c. Lineal Data File Generation

The final phase of the FISS program is devoted to the generation of the final lineal formatted data file. The functions performed include the feature header generation and conversion of the feature data set structure to the required standard MMS tape format.

(1) Feature Header Generation

Due to the nature of feature data set generation, the occurrence of points of maxima and minima may cause a single line continuous feature to

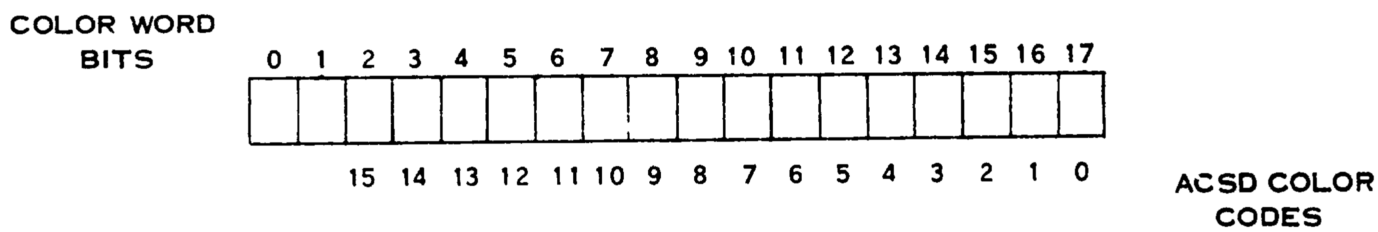


FIGURE 4. FEATURE DATA SET COLOR WORD

be linealized as multiple feature data sets. Because of this, feature identification data must be collected from all the feature data sets which define such a feature to assure proper feature header generation. Color collection involves the accessing and ORing of all feature data set color words which describe a single output feature.

For example, Figure 5 depicts a single closed feature represented as two feature data sets and identified by two unique color tags. Color tag m is correlated to feature data set 1, while color tag n is correlated to feature data set 2. Each feature data set color word, plus the resultant color word following the collection process are depicted.

The resultant color word produced by the collection process is used to create the CCCC field of the MMS feature header record. The CCCC field contains two unique color codes (00_8-20_8). Bits 6-11 represent one color code, while bits 12-17 represent the other color code. In the present configuration, color codes 00_8-17_8 represent the actual ACSD machine codes, while the color code 20_8 is used to represent the color blank (no color).

The base manuscript feature color is contained in the FFFF field of the MMS feature header record. The FFFF field contains a single color code (00_8-17_8) in Bits 12-17, which represents the cartographic feature class, as coded on the base manuscript. Bits 6-11 of the FFFF field of the MMS header record are normally zero unless more than two overlay color tags are correlated to a single output feature, in which case bits 6-11 of the FFFF field are non-zero to indicate possible erroneous color information.

(2) MMS Conversion

During the generation of the lineal data file, the internal disk-resident data set structure is converted to the required ACS MMS format. The PDP-9 fixed point format to GE-645 floating point format conversion is performed by CAST-proven programs, modified to include the feature identification functional capabilities. Logical feature data set linkage functions which concatenate line segment features at points of maxima and minima are also extracted from the CAST software package and implemented in this output processing phase.

d. Statistical Data

The statistical data generated upon successful completion of the FISS is a carry over from the CAST software package. The operational statistics compiled during execution are output via the on-line teleprinter and include the number of points input and output (indication of data expansion or compression, the number of converges and diverges (indication of segmentation and branching), the number of features output, plus storage and processing time information.

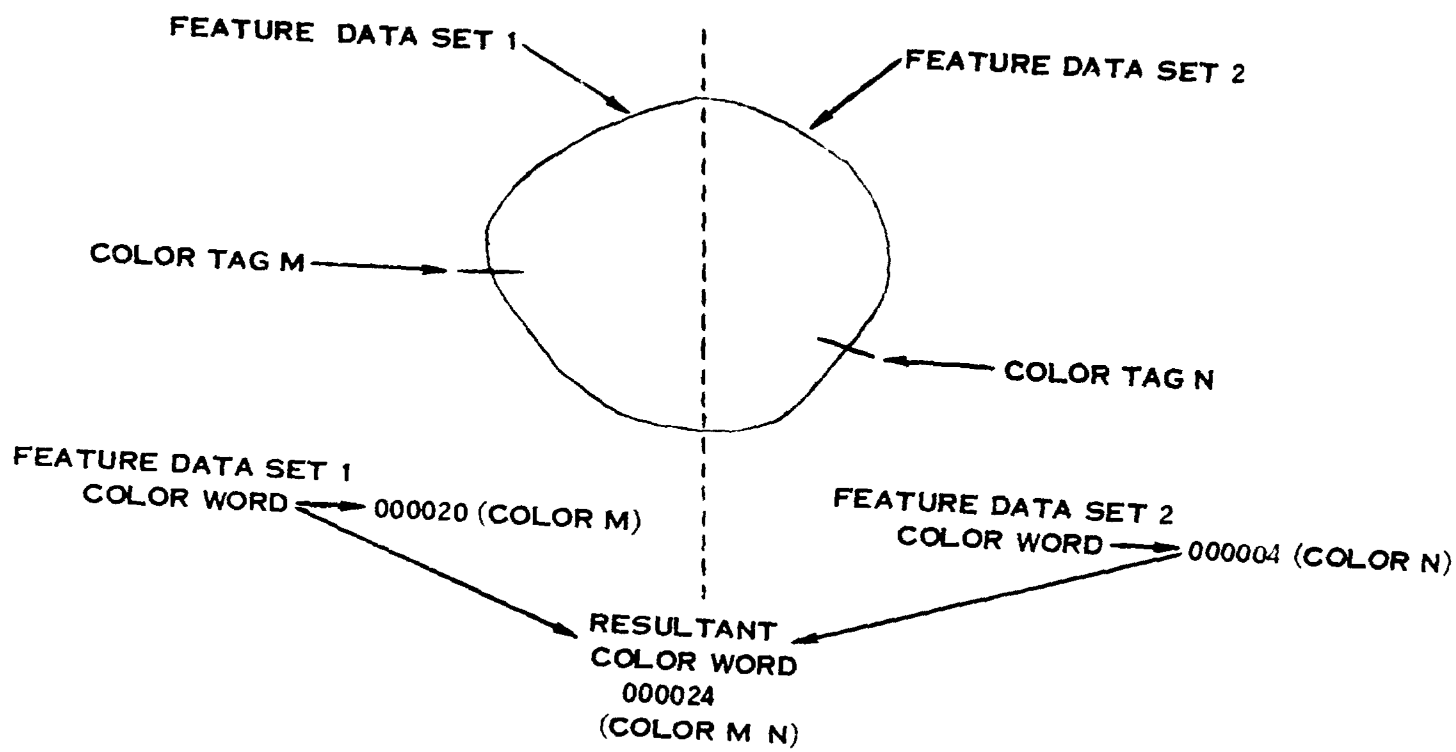


FIGURE 5. COLOR WORD COLLECTION

SECTION IV

RASTER PLOTTER SOFTWARE SYSTEM

1. INTRODUCTION

The Raster Plotter Software System (RPSS) constitutes a set of stand-alone software programs which accept data produced by the Format Conversion Program, and re-structures and formats that data as necessary to drive the ECF Graphic Plotter. The input to RPSS is the edited and symbolized raster formatted data, sorted by the assigned final printed graphics color, as generated by the Format Conversion Program currently in operation at RADC ECF. The outputs from RPSS are the aperture and density level commands required to drive the Graphic Plotter.

2. DESCRIPTION

Since the Raster Plotter Software System accepts data from the Format Conversion Program, and outputs data to the Graphic Plotter, a familiarity with that software and hardware, respectively, is essential prior to describing RPSS itself. The following paragraphs provide functional descriptions of all three systems: Format Conversion, Graphic Plotter, and RPSS.

a. Format Conversion Program

The ACS Format Conversion Program can accept data in either lineal record or raster record format. Lineal record format input is that generated by various lineal devices, or the ACSD with CAST post-processing, and processed by the existing ACS Lineal Processing Software. Raster record format input is that generated directly by the ACSD. For either form of input, the Format Conversion output to RPSS is a raster record format at the final required resolution.

(1) Tape Structure

The Format Conversion Program operates on the GE 635/645; its output magnetic tape file is composed of fixed length blocks which are compatible with the ECF PDP-9 Computer System. The program generates two types of record blocks, each of which contains 320, 36-bit words. These words represent integer values, each GE integer word being equivalent to two PDP-9 integer words.

(2) Data Format

The overall format of these two tape data blocks is shown in Figure 6. Each new scan line begins with a new data block, termed "Start of scan line block". If a scan line contains more data than can be contained within a single 320 word block, then the program generates as many additional blocks as are needed to

Preceding page blank

	WORD	
BLOCK SERIAL NUMBER	1	BLOCK SERIAL NUMBER
RECORD SERIAL NUMBER	2	RECORD SERIAL NUMBER
NUMBER OF X COORDINATES	3	X ADDRESS
FLAG	4	X CONTROL WORD
Y ADDRESS	5	X ADDRESS
Y CONTROL WORD	6	X CONTROL WORD
X ADDRESS	7	X ADDRESS
X CONTROL WORD	8	X CONTROL WORD
X ADDRESS	9	X ADDRESS
X CONTROL WORD	10	X CONTROL WORD
X ADDRESS	11	X ADDRESS
X CONTROL WORD	12	X CONTROL WORD
• • •		• • •
X ADDRESS	119	X ADDRESS
X CONTROL WORD	120	X CONTROL WORD

(a) Start of Scan Block

(b) Continuation Block

Figure 6

Format Conversion Program Block Format

complete the scan line. These additional data blocks are termed "Continuation blocks". Continuation blocks are identical to start of scan line blocks except that they omit the four words designating number of x coordinates, flag, y address, and y command word. These words in the start of scan line block pertain both to that block and to all ensuing continuation blocks.

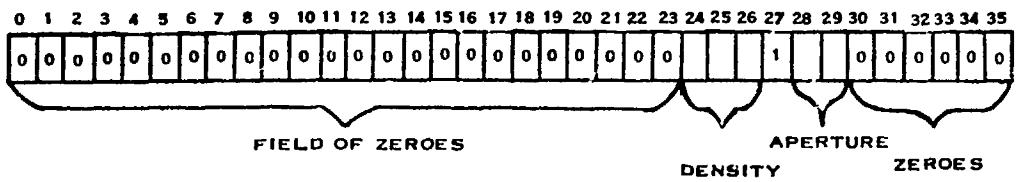
Figure 7 shows the control word associated with the X or Y address given in the proceeding address word. Bits 0 to 23 of the control word are zero, while bits 24 to 29 contain the aperture, density level, and line center designator in the format required to drive the Graphic Plotter. For a lineal format input, bits 30 to 35 are all zero in the control word, while for a raster record format, bits 30 to 35 contain the 5-bit ACSD designator code, right adjusted with one leading zero.

b. Graphic Plotter

The ECF Graphic Plotter is an incremental raster plotting device which uses a modulated laser beam to expose photographic film. The Graphic Plotter is a stand alone unit which receives data from digital magnetic tape and under digital control, plots variable line weight, variable density data. Up to four line weights can be plotted at a single pass, as determined by four preset, selectable apertures, or channels. Area data may also be plotted. For each channel, a density code, with eight specifiable levels, controls the laser beam intensity. The output resolution is selectable at several discrete values up to 1500 lines per inch.

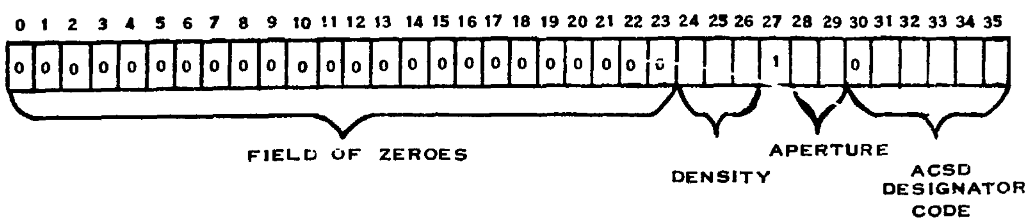
(1) Data Plotting Format

A 24 bit command word controls the operation of the Graphic Plotter. A data block for a single scan line may not contain more than 1024 data words, each word consisting of four, 6-bit characters (see Figure 8). The first character specifies the density/channel selected, while the other three give the scan line address.



(a)

Lineal Record Format

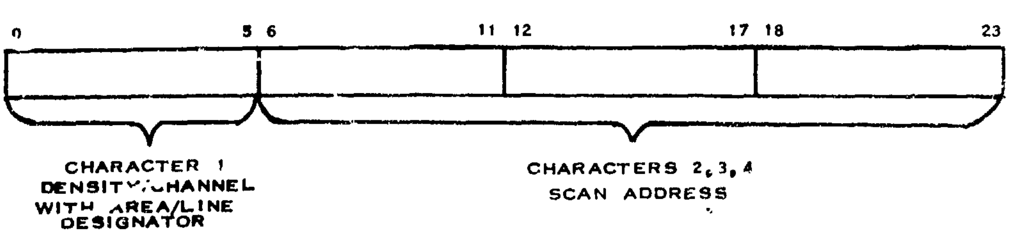


(b)

Raster Record Format

Figure 7

Format Conversion Program Control Word



(c)

Figure 8
Graphic Plotter Command Word

The first word of each block contains the Y-address or carriage position. All subsequent command words contain an X-address or recording data. The bit assignments for the command word are indicated in Table XI for the configuration of the command word shown in Figure 9 . Line weights are achieved by presetting the apertures on each of the four channels. During the run, specific lineweights for each point plotted are effected by selecting one of these channels.

Lineal/areal designation is controlled by using the line/area designator (bit 3) of the graphic plotter command word. If this bit is "1", this signifies that the data point specified by the address in the command word is line center data; this results in a single spot recording. If this bit is "0", this signifies that the address in the command word is an area begin; this results in the laser beam remaining on. Area information is formed by two successive command words, the first containing the area begin designator and the second, to end the area, containing a line center designator and the address at which the beam will be turned off.

(2) Data Termination Format

To terminate a scan line, the command word immediately following the last X-position must contain an end of line (EOL) designator, i.e. bits 7 through 12 of the command word being all 1's. The EOL designator steps the lens carriage to the next scan line. If there is no EOL in the block of command words, the carriage will not be stepped and recording may take place on the same scan line for successive revolutions of the plotter drum. In this case the first command word of the data block must contain the same Y-address, or carriage position, as the previous block.

To terminate recording, an end of message (EOM) signal is indicated in bits 7 through 12 of the command word. In this case bits 7-9 are zero and bits 10-12 are 1's. When an EOM is encountered, recording is stopped. An EOL and an EOM are not to be used in the same data block.

c. Raster Plotter Software System

The function of RPSS is to generate the drive tape required to operate the ECF Graphic Plotter. The raster formatted information used in the genera-

TABLE XI

GRAPHIC PLOTTER WORD FORMAT

BITS POSITION	FIELD DISIGNATION
0-2	Recording Density
3	Line/Area Designator
4, 5	Line Width (Channel)
6	Unassigned
7-23	Scan Position

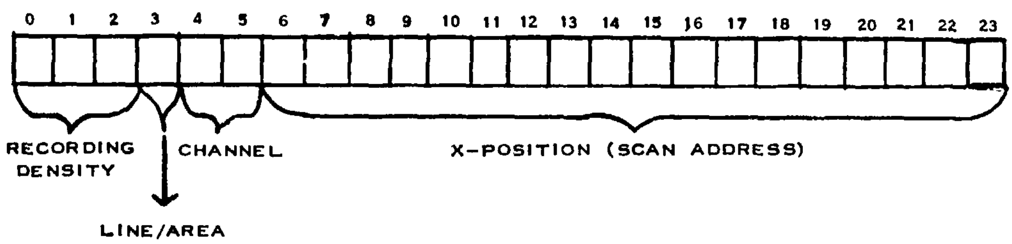


Figure 9
Graphic Plot Command Word Format

tion of plotter commands is obtained from the Format Conversion Program. The required Graphic Plotter command word blocks are output from RPSS and stored on standard 7-level magnetic tape acceptable to the Graphic Plotter.

(1) Design

RPSS is designed to permit the user to exercise a considerable degree of flexibility in generating Graphic Plotter drive tapes. If the user desires to plot all features just as they are represented on the input data tape, the program can be run in Tape Mode. In this mode channel and density assignments are made automatically, based upon the appropriate field in the Format Conversion Program control word. If the user desires to selectively output certain features, or make his own channel density assignments, the program can be run in Manual Mode. In this mode, by using necessary feature selection codes, both feature selection and channel density assignments are made manually. An "ALL ELSE" function permits a common channel/density assignment to be made to all non-selected features.

(2) Implementation

Figure IV, Appendix IV, Section I, shows the overall system block diagram for RPSS. The mode of operation is determined through an operator/program query-response mechanism. When the input magnetic tape data file is read, each address data word and control word is queried. For each address, the proper tag is formulated containing the recording density and channel assignment necessary to effect proper Graphic Plotter operation. When a minimum area is encountered, the proper area begin/area end commands are generated. This procedure is repeated, generating twenty-four bit command words for each scan address.

At scan line completion, RPSS writes the block of data onto the output tape. A continuation block is generated if the data block is filled before the scan line is completed. The word immediately following the last scan address in the output block contains the EOL designator. The EOM designator is used to terminate the Graphic Plotter operation in place of the EOL designator for the last scan line of the manuscript.

3. RPSS OPERATION

Processing of input data requires the assignment of a particular recording density and a particular lineweight to each scan address of the features represented. This may be accomplished in either Tape mode or Manual mode as described below.

a. Tape Mode

(1) Operation

In this mode, automatic channel/density assignment is made based

on the control word contents for each scan address. Initialization of the Tape Mode is described in the Raster Plotter Software System Operating Instructions. (Appendix III, Section II, Paragraph 4). Control information required by the operator for successful operation of this mode is the minimum area in proper resolution. Operation is as shown in the functional block diagram of Figure IV-3.

(2) Data Format

Each input data element consists of the scan address with its associated control word. Upon obtaining this data element, the scan address, channel number and recording density are extracted. A Graphic Plotter command word is then built containing this information. When a minimum area is encountered, an area begin address and an area end address are generated. This process is repeated until either the output buffer becomes filled or the scan line is completed. After the last scan address, an EOL designator is entered and the buffer is written onto the output tape. Processing continues until all input data tapes are completed. When the last scan line is reached, an EOM designator is entered in place of the EOL designator.

After the feature data set has been completed, summary statistics are printed via the on-line printer. Depending on operator response, the program will either recycle to accept new data files, or terminate.

b. Manual Mode

(1) Operation

In this mode, operator designated channel/density assignments are made for selected features represented on the input data tape. Initialization of the Manual Mode is described in the RPSS Operating Instruction (Appendix III, Section II, Paragraph 6). Control information, in addition to minimum area in proper resolution, is obtained by a query-response cycle in order to afford the operator ease in assignment of the desired lineweight and recording densities for feature output. Operation is shown in the functional block diagram of Figure IV-4.

(2) Selection Codes

Selection code entries allow for extraction of single features and ranges of features with consecutive codes. Features can also be suppressed from the output and lineweights and recordeing densities can be reassigned. After the selection codes have been input, they are output on the line printer, allowing corrections to be made if required.

(2) Data Format

Processing begins when the selection code table has been accepted. Each input data element (scan address and control word) is examined for its selection code, located in the same field of the control word as the channel/density designation of the Tape Mode. The selection code table is searched to see if this code is to be output. The data element will be processed if the code is in the table or the ALL ELSE flag is on. When the entry is found in the selection code table, the delete flag is checked. If the delete flag is on, the data element is not processed; otherwise the Graphic Plotter command word is formed for the channel and density previously assigned this code. The ALL ELSE function allows data elements to be processed without the code being in the table. Processing then proceeds in the same manner just described. Minimum area conditions are checked and handled accordingly. Scan lines are terminated with an EOL designator and plotter operation is terminated with an EOM designator.

After the feature data set has been completed summary statistics are printed on the line-printer. Depending on operator response, the program will either recycle to accept new data files, or terminate.

SECTION V

TEST AND DEMONSTRATION

This section describes the tests performed by Hamilton Standard personnel at the RADC ECF for the purpose of demonstrating the performance of the Feature Identification Software System and Raster Plotter Software System. All tests were performed using the DEC PDP-9 computer processing system located at the RADC ECF. The specific tests performed, their results, and the end evaluation of these results are described for both stand alone software systems.

1. FEATURE IDENTIFICATION SOFTWARE SYSTEM (FISS)

a. Method of Approach

Two major series of tests were performed at RADC using hand-drawn graphics to demonstrate the application of the color coding scheme and the performance capabilities of the Feature Identification Software System. The first series of tests used a simple, single-color base manuscript and its associated multi-color coded overlay. The graphic consisted primarily of closed features and maxima-minima features representing various color tag ordering and placement as shown in Figure 10. Such feature data enabled the checkout and evaluation of color tag correlation and collection functional capabilities. The second series of tests used a multi-color base manuscript and its associated multi-color coded overlay. The graphic consisted of typical cartographic features classified by feature color.

All tests were performed on all-areal ACSD-generated data files using nominal CAST line connection software parameters. PDP-9 printer plots were generated to evaluate ACSD raster data inputs to the FISS and to verify raw data anomalies. The CDP and its associated software system were used to plot and selectively verify proper color tag correlation and feature header identification.

b. Test Procedures

The testing procedures are presented in two parts. The first part describes the procedures used to generate the ACSD raster files which serve as test files for the FISS. The second part describes the procedures used to demonstrate the functional capabilities of the FISS.

(1) Test File Generation

(a) Graphic Preparation

Preceding page blank

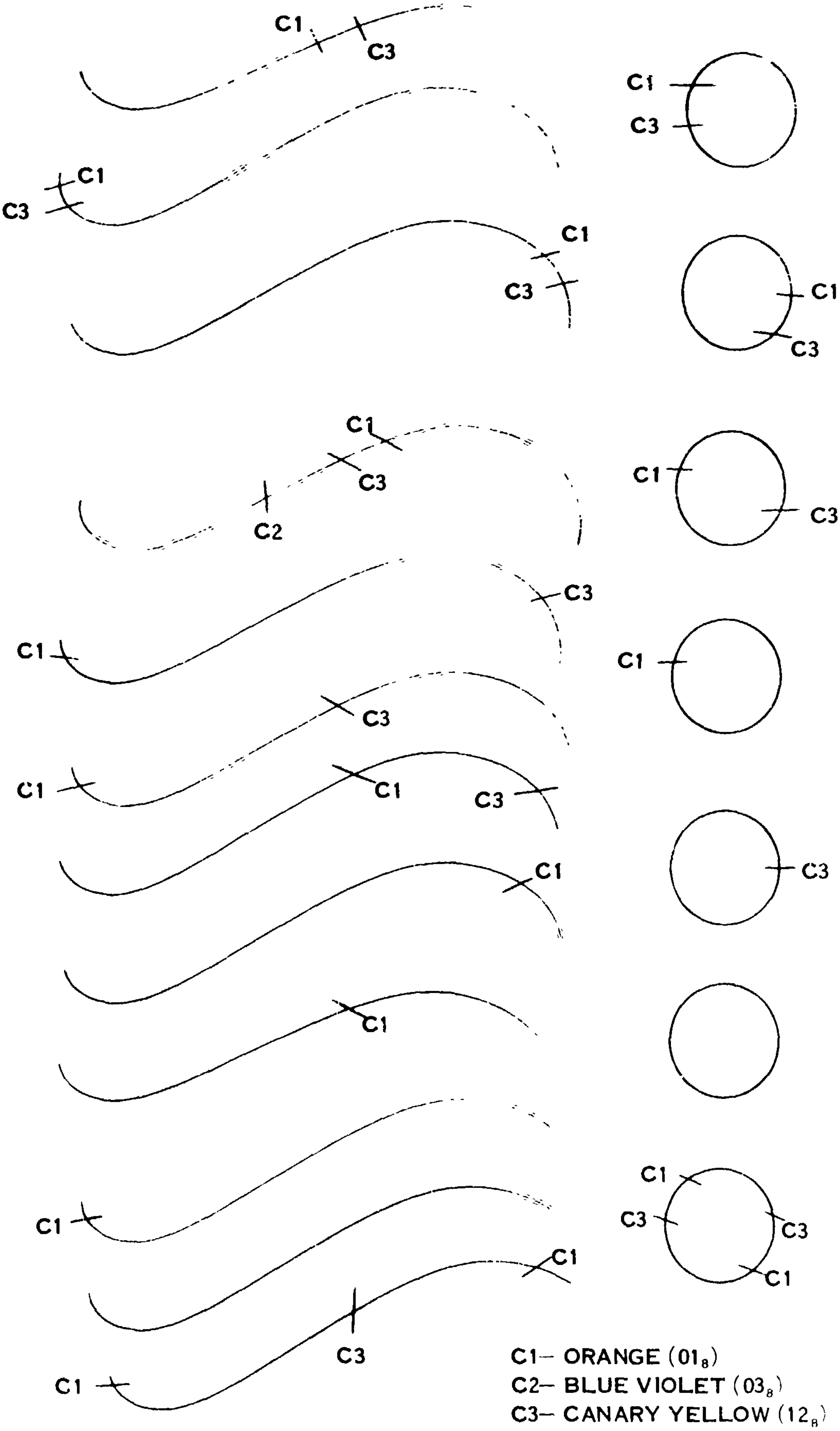


FIGURE 10. FISS SIMPLIFIED TEST GRAPHIC

Two hand-drawn, multi-color graphics were prepared in accordance with RADC operational procedures. These graphics consisted of:

- A hand-drawn base manuscript containing three (3) feature types,(contours, drains, roads)represented by four (4) unique colors.
- A hand-drawn coded overlay containing color tag information represented by a set of seven (7) unique colors plus blank.

(b) Registration

The base manuscript and coded overlay were prepared and scanned using previously described pin registration procedures such that the information derived from successive scans of base and overlay produced properly registered raster data file.

(c) - Machine Parameter Settings

The base manuscript and coded overlay were scanned successively using a minimum area setting of one and a resolution of four (4) mils. The resultant test files were a base manuscript file and a coded overlay file, each on a separate tape.

(2) FISS Execution

The Feature Identification Software System produced MMS formatted lineal feature files, one for each feature type represented by a base manuscript color. The FFFF field of each lineal feature should record the base manuscript color code while the CCCC field should record the color code representing the color tags on the coded overlay.

c. Test Results

The CDP and its associated software system were used to evaluate the feature identification capabilities of the FISS. The FFFF field of each lineal feature header identifies the base manuscript color of the feature type. In addition, when the leading pair of digits of the FFFF field of the lineal feature header are non-zero, a flag is set to indicate possible erroneous color tag information, as created by the correlation of more than two unique color tags to one lineal feature. The CCCC field is used to identify the overlay color tags associated with the lineal feature. Selective examination of lineal features and their associated headers verified that the prescribed color tag correlation and identification had taken place.

(1) Initial Testing

The initial series of tests subjected the FISS to closed features and maxima-minima features representing various color tag orders and placements as shown in Figure 10. Color tag correlation and collection were successful and proper feature identification resulted for the various color tag orientations presented on the test graphic. Color tag correlation and proper feature identification resulted for all no-, one-, and two-color tag representations. The single more-than-two-color tag case was properly recognized and flagged for possible erroneous color information. Redundant color tag coding (overcoding) had no effect on the correlation, collection, or identification functions of the FISS. The resultant feature headers generated are shown in Table XII.

FFFF	CCCC	Color 2	Color 1
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
7711	Ø7Ø1	Blue Violet	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	2ØØ1	Blank	Orange
ØØ11	2ØØ1	Blank	Orange
ØØ11	2ØØ1	Blank	Orange
ØØ11	2Ø2Ø	Blank	Blank
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	12Ø1	Canary Yellow	Orange
ØØ11	2ØØ1	Blank	Orange
ØØ11	2Ø12	Blank	Canary Yellow
ØØ11	2Ø2Ø	Blank	Blank
ØØ11	12Ø1	Canary Yellow	Orange

TABLE XII

FISS FEATURE HEADERS: INITIAL TESTING

(2) Final Testing

The second series of tests subjected the FISS to typical cartographic features (contours, drains, roads). Table XIII shows the set of feature header identification fields generated by the FISS for the base manuscript color representing drainage. Also presented in Table XIII is the actual color identification represented on the base manuscript and coded overlay describing these drainage features.

Table XIII indicates a relatively high occurrence of the no-color tag representation. It was verified during FISS testing that this no-color tag representation was attributable to non-tagged features caused by feature segmentation and/or miscorrelation of color tags due to scan line shift resulting from the data loss in the overlay file. The ACSD file data loss was verified by PDP-9 line printer plots during FISS testing. Table XIII also indicates proper one-color tag representations for both magenta and canary yellow color tags.

There were two occurrences of improper feature identification other than the no-color tag case. The first one was a valid two-color tag representation, but an erroneous feature identification. The second occurrence was an invalid feature identification of more than two unique color tags. It was verified, however, using the CDP Software System that both these occurrences were attributable to the previously mentioned data loss, and not related to FISS software deficiencies.

d. Evaluation of Test Results

The intent of the Feature Identification Software System was to implement the color coding scheme prescribed by the study and investigation phase (Section II). Implementation involved color tag correlation and feature identification, both of which were successfully accomplished by the FISS. The following paragraphs disc [illegible] particular areas of interest.

The ACSD pin registration retrofit proved adequate to produce base manuscript and coded overlay files having sufficiently accurate registration to permit successful implementation of the proposed coding scheme. No software registration was required.

Although the current version of the FISS requires a fixed resolution for successive scans of base and overlay manuscripts, a decrease in scanning time could be obtained by scanning the coded overlay at a coarser resolution than the base manuscript. With minor software modifications, this option could be incorporated into the FISS.

ACTUAL COLOR
TAG IDENTIFICATION

BASE FEATURE	BASE COLOR	COLOR TAG 2	COLOR TAG 1
DRAINS	BLUE VIOLET(Ø7_8)	BLANK(2$Ø_8$)	MAGENTA(Ø3_8)
DRAINS	BLUE VIOLET(Ø7_8)	BLANK(2$Ø_8$)	CANARY YELLOW (1$2_8$)

RESULTANT COLOR
TAG IDENTIFICATION

NO. OF FEATURES	BASE COLOR $(FFFF)_8$	NO. OF COLOR TAGS	COLOR TAGS2/1 $(CCCC)_8$	COLOR2	COLOR1
11	ØØØ7	0	2Ø2Ø	Blank	Blank
8	ØØØ7	1	2ØØ3	Blank	Magenta
9	ØØØ7	1	2Ø12	Blank	Canary Yellow
1	ØØØ7	2	12Ø3	Canary Yellow	Magenta
1	77Ø7	>2	1ØØ1	Azure Blue	Orange

TABLE XIII

FISS FEATURE HEADER: FINAL TESTING

Because the coding scheme does not utilize color repeats in the overlay code structure, the use of redundant color tagging was shown to have no adverse effects on FISS operation. Moreover, such redundant coding was shown to be both useful and necessary in perpetuating the proper feature identification across unintended gaps in the CAST-generated lineal data strings. Otherwise, such feature segments would become associated with the default header, i.e., a one-color or no-color tag.

Although only slash lines were used as color tags in the tests performed to date, it should be emphasized that the FISS requires no special code shape; the only requirement is for near intersection of feature and overlay symbol. It is therefore suggested that testing be conducted to determine what other symbol variations are convenient to the cartographer, provide reasonable redundancy, and result in the lowest error rates.

2. RASTER PLOTTER SOFTWARE SYSTEM (RPSS)

a. Method of Approach

Two major series of tests were performed to demonstrate the two different modes of operation available using the RPSS. The first series of tests were performed using the Tape Mode of operation, while the second series were performed using the Manual Mode. Both series of tests were performed on the same test data file in order to better verify and compare results. The test data file consisted of a cartographic manuscript which had been digitized and subsequently processed by the Format Conversion Program for use by the RPSS.

Once the output tapes for each test were generated, they were plotted on the ECF Graphic Plotter for final evaluation. Tape dumps using the utility routine MTDUMP and the PDP-9 line printer were also made of both the input test data file and of each output plot tape to test the accuracy and proper operation of the RPSS program.

b. Test Procedures

Separate tests were performed for each mode of operation, Tape Mode and Manual Mode. The paragraphs that follow first describe the test to demonstrate the Tape Mode of operation, and second, the tests to demonstrate the Manual Mode of operation.

(1) Tape Mode

The program was initialized for operation in the Tape Mode in accordance with the instructions provided in the RPSS operations manual (Apppendix III). The prescribed test input records were converted from their 36-bit raster record data word format to the prescribed Graphic Plotter 24-bit raster record data word format. The converted data was assembled and recorded on magnetic tape in the block record format of the Graphic Plotter. Control data was incorporated within the output data stream as required to identify the start and end of scan line plot records, beginning and end of areas, aperture and density parameters, etc. All test data appearing in the test input data file was processed and recorded on the test output file.

(2) Manual Mode

This series of tests was conducted to demonstrate the versatility and feature extraction provisions of the RPSS. The tests allowed for selectability of features desired to be plotted.

(a) Single Feature Pull Plot Test

The term "Single Feature Pull" as used in the context of this test description is defined as all cartographic information appearing in the test file which is tagged with the same selection code designator. This test demonstrated the mechanism by which such information was identified and extracted from the test input file and then processed and recorded so as to create a "single feature" Graphics Plotter input tape.

(b) Multiple Feature Pull Plot Test

This test demonstrated the ability of the RPSS to map two or more features, defined in the same context as the previous test, to any legitimate user specified Graphics Plotter output symbolization. This was accomplished by assigning two features to one output channel/density, and a third feature to a different output channel/density.

(c) All Else Test

This test was included to demonstrate the "All Else" provisions of the RPSS. This provision may be invoked at any time during the program initialization sequence. Its function is to provide the user with the capability to plot all remaining unassigned input data at a predefined Graphics Plotter aperture and/or density setting.

c. Test Results

Test output verification was obtained in two forms. The first form was the plotted hardcopy output from the ECF Graphic Plotter. The second form consisted of line printer dumps obtained from both input and output tapes. Figures 11 and 12 show the hardcopy output of the Tape Mode and a single feature pull of the Manual Mode, respectively. Figures 13 and 14 show a typical tape dump of the input and output tapes respectively.

All tests were processed to completion. Four output tapes were generated, one for each of the tests described in paragraph 2.b above. Tape dumps were obtained for the input tape data file and each of the four output tapes. The dumps were taken in the same vicinity in order to show correlation and proper processing of the data for each test. Graphic Plotter output was not generated initially due to operational difficulties encountered with that device. Consequently, evaluations were first made using the tape dumps. When the Graphics Plotter became operational, plotted confirmation of the lineprinter tabulations was obtained. The following paragraph discusses the evaluation as based on all available test results.

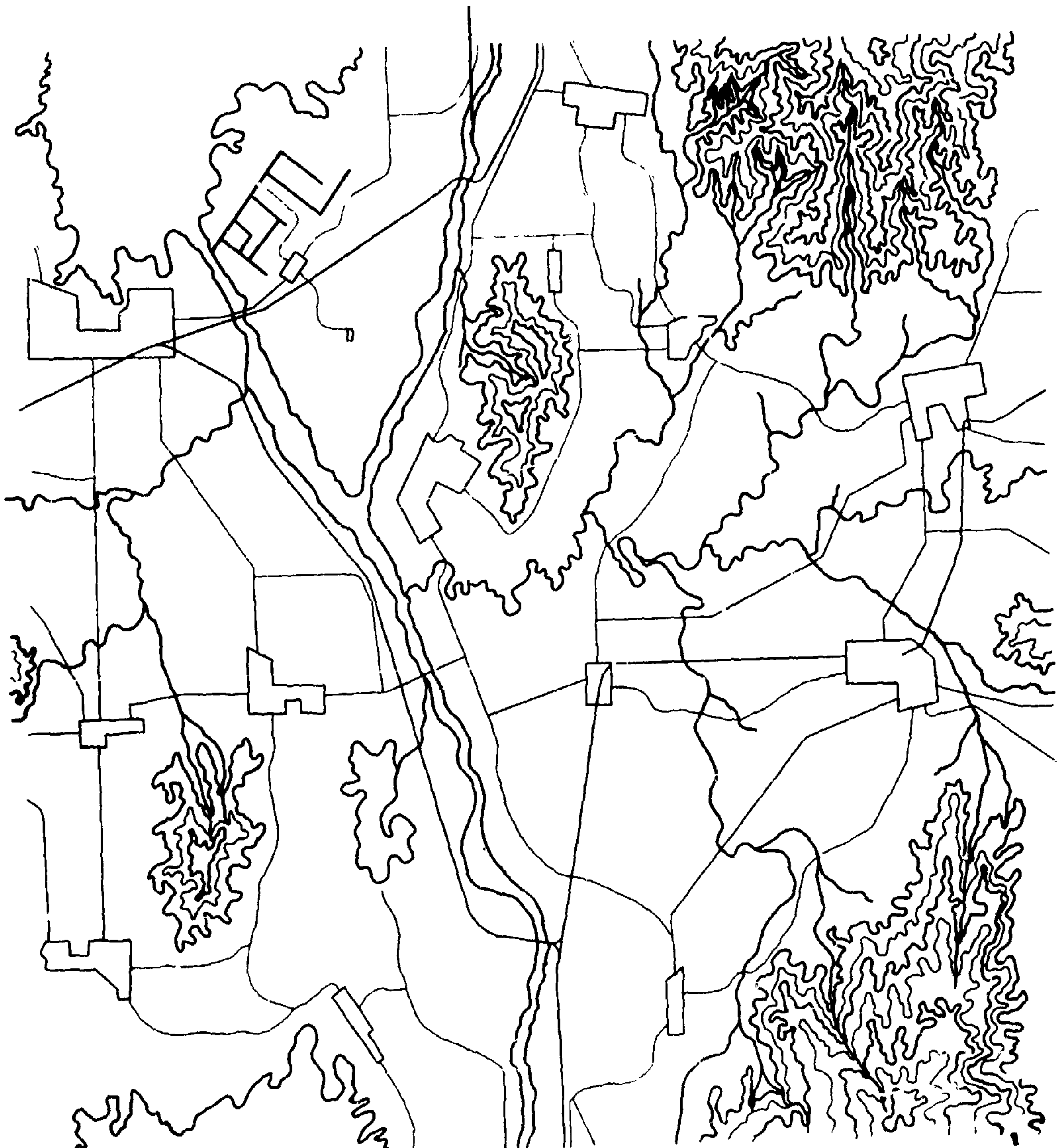


FIGURE 11. SAMPLE GRAPHIC PLOTTER OUTPUT: TAPE MODE

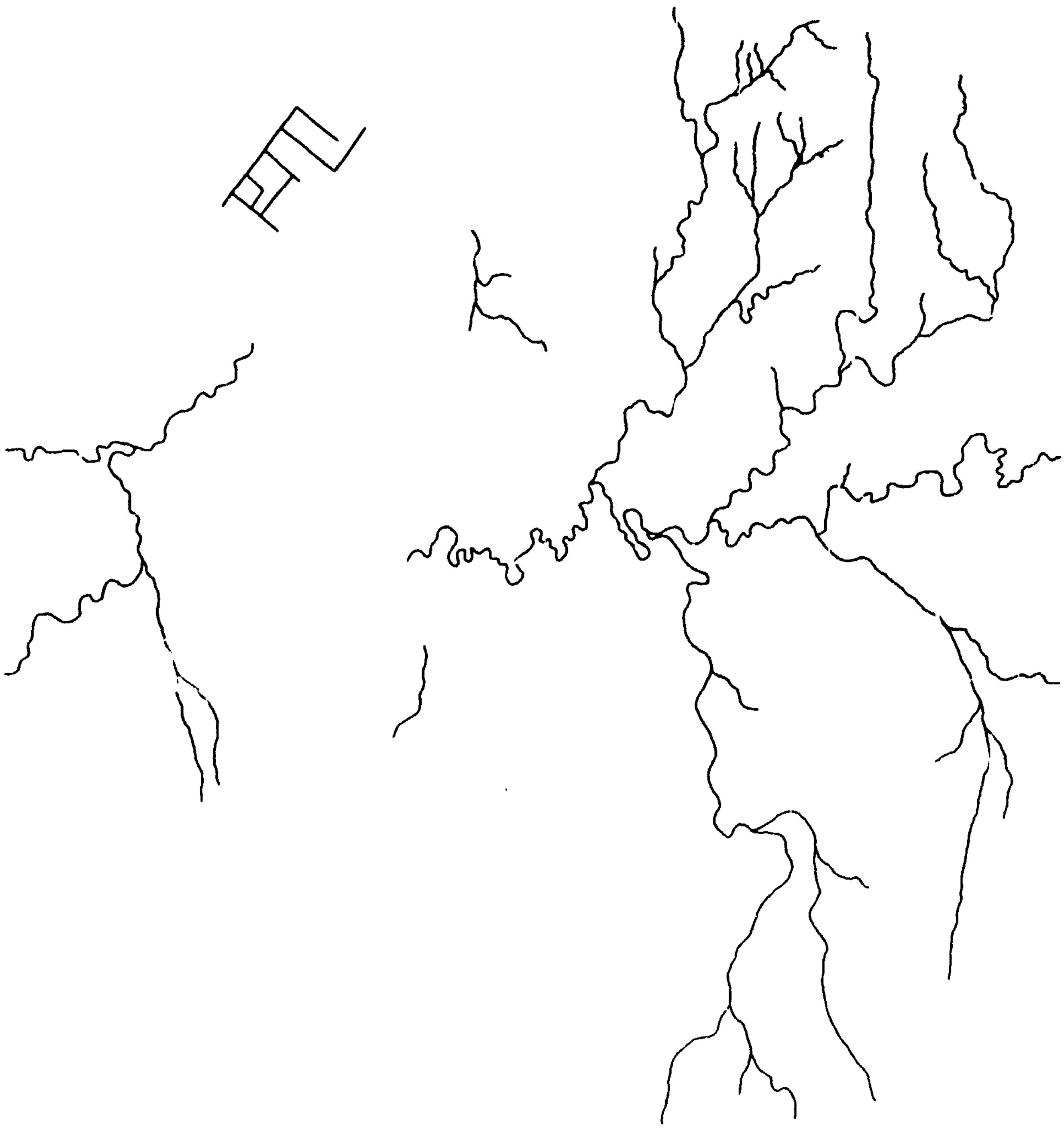


FIGURE 12. SAMPLE GRAPHIC PLOTTER OUTPUT: MANUAL MODE

WORD NUMBER	BLOCK SERIAL NUMBER		RECORD SERIAL NUMBER		NUMBER OF X COORDINATES		FLAG OF 6'S		
1	032760	000477	000476	000000	000000	000044	666666	666666	000000
10	002752	[illegible]	000010	000030	000342	000000	005604	000000	000377
19	000000	007504			X ADDRESS		X CONTROL WORD		X ADDRESS
28	X CONTROL WORD		Y CONTROL WORD		007504	000000	000603	000000	007604
					000000	000732	000000	005504	000000
Y ADDRESS									
		[illegible]	005704	000000	001236	000000	005504	000000	002041
55	000000	007604	000000	002322	000000	005504	000000	002416	000000
64	005504	000000	003047	000000	003704	000000	003162	000000	003704
73	000000	003163	000000	003704	000000	003164	000000	003704	000000
82	003165	000000	003704	000000	003176	000000	007704	000000	003236
91	000000	007744	000000	003327	000000	005704	000000	003374	000000
100	003604	000000	004000	000000	005504	000000	004073	000000	005504
109	000000	004374	000000	007604	000000	004641	000000	003604	000000
118	004732	000000	003604	000000	005064	000000	007604	000000	005104
127	000000	007504	000000	005144	000000	005604	000000	005147	000000
136	005604	000000	005150	000000	005604	000000	005151	000000	
145				005604	000000	005153	000000	005604	000000
154	005154	000000	005604	000000	002345	000000	005504	000000	
	X ADDRESS	X CONTROL WORD		002347	000000	005504	000000	002350	000000
172	005504	000000	002351	000000	005504	000000	002352	000000	005504
181	000000	002353	000000	005504	000000	002354	000000	005504	000000
190	002355	000000	005504	000000	002356	000000	005504	000000	002357
199	000000	005504	000000	002360	000000	005504	000000	002361	000000
208	005504	000000	002362	000000	005504	000000	002363	000000	005504
217	000000	002364	000000	005504	000000	002365	000000	005504	000000
226	002366	000000	005504	000000	002367	000000	005504	000000	002370
235	000000	005504	000000	002371	000000	005504	000000	002372	000000
244	005504	000000	002373	000000	005504	000000	002374	000000	005504
253	000000	002375	000000	005504	000000	002376	000000	005504	000000
262	002377	000000	005504	000000	002400	000000	005504	000000	002401
271	000000	005504	000000	002402	000000	005504	000000	002403	000000
280	005504	000000	002404	000000	005504	000000	002405	000000	005504
289	000000	002406	000000	005504	000000	002407	000000	005504	000000
298	002410	000000	005504	000000	002411	000000	005504	000000	002412
307	000000	005504	000000	002413	000000	005504	000000	002414	000000
316	005504	000000	002415	000000	005504	000000	002416	000000	005504
325	000000	003047	000000	003704	000000	003160	000000	003704	000000
334	003161	000000	003704	000000	003162	000000	003704	000000	003177
343	000000	007704	000000	003236	000000	007704	000000	003327	000000
352	005774	000000	003374	000000	003604	000000	004000	000000	005504
361	000000	004073	000000	005504	000000	004374	000000	007604	000000

TAPE DUMP – INPUT TAPE DATA FILE

FIGURE 13. LINEPRINTER DUMP:. RPSS INPUT DATA

	Y COMMAND WORD	X COMMAND WORD	X COMMAND WORD		X COMMAND WORD	X COMMAND WORD	X COMMAND WORD		
1	040727	525[illegible]	034275	000377	750004	057500	040675	000473	
10									
19	241637	[illegible]	370031	623700	316337	003164	370031	657700	317677
28	003236	[illegible]	273600	337455	004000	550040	737600	437436	004641
37	360047	327600	506475					005150	560051
46	5156[illegible]	515256	005153	560051	540437	400004	370000	043700	000437
55	000004	370000	043700	X COMMAND WORD	END OF LINE DESIGNATOR		HIGH IGNORE VALUE	000437	000004
64	370000	043700	000437					000004	370000
73	043700	000437	000004	370000	043700	000437	000004	370000	043700
82	000437	000004	370000	043700	000437	000004	370000	043700	000437
91	000004	370000	043700	000437	000004	370000	043700	000437	000004
100	370000	043700	000437	000004	370000	043700	000437	000004	370000
109	043700	000437	000004	370000	043700	000437	000004	370000	043700
118	000437	000004	370000	043700	000437	000004	370000	043700	000437
127	000004	370000	043700	000437	000004	370000	043700	000437	000004
136	370000	043700	000437	000004	370000	043700	000437	000004	370000
145	043700	000437	000004	370000	043700	000437	000004	370000	043700
154	000437	000004	370000	043700	000437	000004	370000	043700	000437
163	000004	370000	043700	000437	000004	370000	043700	000437	000004
172	370000	043700	000437	000004	370000	043700	000437	000004	370000
181	043700	000437	000004	370000	043700	000437	000004	370000	043700
190	000437	000004	370000	043700	000437	000004	370000	043700	000437
199	000004	370000	043700	000437	000004	370000	043700	000437	000004
208	370000	043700	000437	000004	370000	043700	000437	000004	370000
217	043700	000437	000004	370000	043700	000437	000004	370000	043700
226	000437	000004	370000	043700	000437	000004	370000	043700	000437
235	000004	370000	043700	000437	000004	370000	043700	000437	000004
244	370000	043700	000437	000004	370000	043700	000437	000004	370000
253	043700	000437	000004	370000	043700	000437	000004	370000	043700
262	000437	000004	370000	043700	000437	000004	370000	043700	000437
271	000004	370000	043700	000437	000004	370000	043700	000437	000004
280	370000	043700	000437	000004	370000	043700	000437	000004	370000
289	043700	000437	000004	370000	043700	000437	000004	370000	043700
298	000437	000004	370000	043700	000437	000004	370000	043700	000437
307	000004	370000	043700	000437	000004	370000	043700	000437	000004
316	370000	043700	000437	000004	370000	043700	000437	000004	370000
325	043700	000437	000004	370000	043700	000437	000004	370000	043700
334	000437	000004	370000	043700	000437	000004	370000	043700	000437
343	000004	370000	043700	000437	000004	370000	043700	000437	000004
352	370000	043700	000437	000004	370000	043700	000437	000004	370000
361	043700	000437	000004	370000	043700	000437	000004	370000	043700

TAPE DUMP – OUTPUT TAPE DATA FILE

FIGURE 14. LINEPRINTER DUMP: RPSS OUTPUT DATA

d. Evaluation of Test Results

In reviewing the hard copy output of the Graphic Plotter, it was noted that some scan lines, or portions of scan lines, were missing. An examination of input tape dumps, taken in the vicinity of the missing scan lines, showed that the data was there and had been properly processed. An examination of output tape dumps taken in the same vicinity however, showed that the Y-scan address had been improperly processed. Upon HSD recommendation, the input tape data file was regenerated by RADC using new, unused magnetic tape. The test was then rerun under the same conditions as before, using new tape for output as well. When plotted on the Graphic Plotter, no scan lines were missing. This confirmed that the RPSS program was operating correctly. The original source of error was not definitely isolated although a likely cause was the difference between the magnetic tape units of the GE 635/645, where the input tape data file was generated, and those of the DEC PDP-9, where the input tape data file was read for processing.

To verify the RPSS Program processing, it was necessary to examine the tape dumps taken at the completion of the tests. The dump shown in Figure 13 shows the input tape dump. An input record is 320 words long, each word having 36 bits. This corresponds to a block of 640 PDP-9 words, each word having 18 bits. For convenience in reading the dump, "words" hereafter refer to 18-bit PDP-9 words.

As indicated in the figure, words 1 and 2 contain the block serial number; 3 and 4 contain the record serial number; 5 and 6 contain the number of X coordinate; 7 and 8 contain a flag of 6's; 9 and 10 contain the Y address; 11 and 12 contain the Y control word; 13 and 14 contain the X address; 15 and 16 contain the X control word. The X address and X control word are repeated until the block is finished. The X control word contains the channel number and density in proper format to drive the Graphic Plotter. The control word also contains the ACSD designator code. For example, observe word number 16 in the dump, which is 005604. The 56 is the octal representation of the channel and density, 101110 in binary. This means density of 5, channel 2, line center data. The 04 is the octal representation of the ACSD designator code, which in this case signifies line center data. It is data of this format that is converted by the RPSS Program. The control word follows its associated address. In this case word 16 is the control word for the address appearing in word 14, i.e. 000342 (octal representation).

The RPSS Program converts this data to the 24-bit word representation required to drive the Graphic Plotter. An output tape dump is shown in Figure 14. The Y- and X- command words are indicated, each being eight octal characters in length. Since the PDP-9 word is 18-bits (6 octal characters), the dump representation shows the command word split. For the address and control word discussed above, the Graphic Plotter command word is found in words 2 and 3 of the output tape dump, i.e., the last four characters of word 2 and the first four characters of word 3. To determine the scan line on the Graphic Plotter, an end-of-line-designator is required. This is shown in words 50 and 51 of the output tape dump

as the octal representation 04734000. The remainder of the buffer is fielded with an address higher than any acceptable scan address and thus will be ignored by the Graphic Plotter.

Based on the hardcopy output of the Graphic Plotter, as augmented by the line- printer dump verifications, it is concluded that the RPSS Program meets all design requirements and generates the necessary tapes for the ECF Graphic Plotter.

SECTION VI

CONCLUSIONS AND RECOMMENDATIONS

The objectives of this program were to expand the capabilities of the Raster Processing Subsystem with regards to both its Raster Data Input and its Raster Data Output. These objectives have been accomplished and are described below along with their related conclusions and recommendations.

1. RASTER DATA INPUT

A color coding scheme for the raster data recorded by the ACSD has been developed and implemented. The purpose of this scheme is to extend the present feature class identification capability to the level necessary to hierarchially identify all classes, subclasses, and types of features as found on JOG 1501-A series charts.

Initial testing with this color coding scheme, as embodied in the Feature Identification Software System (FISS), confirms that these basic performance objectives have been satisfactorily attained. The code devised is easy for the compiler to apply and permits conventional hardware detection. Further, it is straightforward for the software to assimilate and associate with the proper line form data.

The only noteworthy shortcoming of the coding scheme, as implemented, is its limited ability to perpetuate code symbology along the entire lengths of complex lineal features. This shortcoming is directly attributable to known limitations within the current raster-to-lineal conversion, or line connect algorithms. These limitations cause certain types of complex lineal data to "segment" into a number of unrelated data sets. It is this segmentation which impedes the complete code-to-feature correlation which is essential to proper color code identification.

In devising the color code strategy, it was recognized that -- while outside the scope of this effort -- an improved line connect capability would ultimately be required within the Raster Processing Subsystem. The decision was therefore made to accept the necessity of coding individual feature segments, i.e. use redundant coding, on the proviso that, as the line connect capability improved, the degree of redundancy needed in utilizing the code would likewise diminish.

The recommendation is therefore made that the necessary efforts be taken to further advance the current line connect capability employed by the CAST software on the PDP-9 computer facility. Such advancements would significantly improve current ACS operation and bring the newly devised color code identification scheme to fruition.

Preceding page blank

2. RASTER DATA OUTPUT

Software for symbolizing raster data for output on the Graphic Plotter has been developed and implemented. The purpose of these programs is to convert existing linecenter formatted raster data to the specific lineweight and density command codes required to operate that plotting device.

Initial testing with this raster output symbolization software, as embodied in the Raster Plotter Software System (RPSS), confirms that these basic performance objectives have also been satisfactorily attained. The software features both an automatic Tape Mode, as would be used for outputting fully edited and completely preformatted data, and a Manual Mode, which permits the operator to make specific data selections and format alterations.

The only shortcomings observed were "missing data" errors which were ultimately attributed to tape unit differences among the GE-635/645 drives used to write the input tape for RPSS, the PDP-9 drives used to read the input tape and prepare the output tape during RPSS execution, and the Graphic Plotter drive used to read the output tape for plotting. Such incompatibilities among supposedly "IBM Compatible" systems are not uncommon. A standard Master Alignment tape, plus a proven pair of RPSS input and output tapes, should be made available to ECF systems maintenance personnel to facilitate troubleshooting these errors in the future.